

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación



Diseño y evaluación de un protocolo de comunicación de bajo consumo energético para una red inalámbrica de sensores para la monitorización ambiental

Tesis sometida a consideración de la Escuela de Ingeniería en Computación para optar por el grado de magister Scientiae en computación con énfasis en ciencias de la computación

Ronny Giovanni García Ramírez

Cartago, noviembre, 2013

Capítulo 1

Resumen

El presente proyecto consiste en el diseño del protocolo de comunicación utilizado en una red de sensores para la monitorización de variables diversas en el bosque; el protocolo abarca funciones correspondientes a las capas de red, control de acceso al medio y de enlace de datos según el modelo de interconexión de sistemas abiertos (OSI).

Entre los principales problemas que se enfrentan en el presente proyecto se encuentran la reducción del consumo energético del módulo de comunicación con respecto del protocolo IEEE802.15.4, obtener una latencia aceptable en la transmisión de la información y lograr que cada nodo de la red tenga la capacidad de configurarse por sí mismo de manera que los módulos no tengan que ser modificados en procesos de expansión o falla.

Se propone un protocolo multicapa para la transmisión de alarmas desde los nodos de la red hacia nodos base o sumideros, en el que se minimiza el número de saltos de las tramas en la capa de red, y se agrupan los datos con el objetivo de reducir las colisiones en la red y de esta manera reducir el desperdicio energético asociado a dichas colisiones. Además se utilizan ciclos en los que se apaga la etapa de transmisión-recepción de los nodos de la red y se modela la mejora en el consumo de la red con respecto de un protocolo que mantiene la etapa de comunicación siempre encendida.

El protocolo propuesto es evaluado con respecto al consumo de potencia usando como referencia la interfaz de red de Texas Instruments CC2420 la cual usa el estándar IEEE802.15.4, mismo estándar usado por zigbee, además se usa wiseroute como protocolo de red. Para hacer la simulación se utiliza Omnet++ y la biblioteca de simulación Mixim. Se comprueba la tolerancia a cambios en la red del protocolo por medio de pruebas. Se realizan además pruebas de consumo de potencia y de latencia de las alarmas desde que se producen hasta que llegan a la base con diferentes niveles de tráfico en la red y variando el ciclo de trabajo de los nodos, se comprueba que el protocolo propuesto es capaz de proveer aproximadamente un 90% de ahorro energético con respecto de los nodos de referencia usados, además se comprueba que la red es capaz de mantener una latencia promedio de las alarmas por debajo de 20 minutos para alarmas que se generan a 50 niveles de distancia de la base.

Abstract

This project consists on the design of the communication protocol used in a wireless sensor network to monitor variables in the forest; it covers protocol layer functions corresponding to the network, medium access control and data link layer according to the model of open systems interconnection (OSI) .

Among the main problems faced in this project it is the reduction in the power consumed by the nodes compared with a node which implements IEEE802.15.4, obtain an acceptable latency in the transmission of information and to ensure that each network node has the ability to configure itself so that the modules do not have to be manually modified in the event of expansion or fails.

A multilayered protocol for the transmission of alarms from network nodes to sink nodes is proposed. This protocol minimizes the number of hops for the frames in the network layer and uses data grouping to reduce network collisions and thereby reduce the energy waste associated with such collisions. Cycles where the transmission-reception module of the nodes is turned off are used and the improvement in power consumption due to such cycles is modeled and compared against a node which don't use those turn-off cycles. The proposed protocol is evaluated for power consumption using as reference the network interface Texas Instruments CC2420 which uses the standard IEEE802.15.4, same standard used by ZigBee; for the reference node wiseroute was used as network protocol . The simulations are made using Omnet++ and the Mixim library. The tolerance of the protocol to changes is proven by tests designed using Omnet++. Tests are also carried out for power consumption and latency of the alarms from their occurrence at different levels of the network until they reach the base at different traffic and network levels while varying the duty cycle of the nodes. It is found that the proposed protocol is capable to provide approximately 90% energy savings in comparison to the reference nodes used, it is also capable of maintaining a mean latency of the alarms under 20 minutes for alarms that are generated 50 levels away from the sink nodes

APROBACIÓN DE LA TESIS

**"Diseño y Evaluación de un Protocolo de
Comunicación de Bajo Consumo Energético para una
Red Inalámbrica de Sensores para la Monitorización
Ambiental"**

TRIBUNAL EXAMINADOR


Ph.D. Pablo Alvarado Moya
Profesor Asesor


M.Sc. Anibal Coto Cortés
Profesor Lector


Ing. Faustino Montes de Oca
Profesor Externo


M.Sc. Luis Carlos Loaiza Canet
Coordinador a.i. del Programa
de Maestría en Computación

a mis queridos Padres

Agradecimientos

El resultado de este trabajo no hubiese sido posible sin los valiosos aportes del Doctor Ing José Pablo Alvarado Moya, quien sacó tiempo de su apretada agenda para ser tutor de esta tesis.

Ronny Giovanni García Ramírez

Cartago, 7 de diciembre de 2013

Índice general

1	Resumen	c
	Índice de figuras	iii
	Índice de tablas	v
2	Introducción	1
2.1	Objetivos y estructura del trabajo	2
3	Marco teórico	5
3.1	Métodos de Ahorro de energía en redes de sensores	5
3.2	Otros protocolos de control de acceso al medio de bajo consumo energético para redes de sensores	10
3.3	Protocolos de red de bajo consumo energético para redes de sensores	11
3.4	Simuladores de redes	13
4	Diseño del protocolo	19
4.1	Descripción del protocolo de control de acceso al medio propuesto	20
4.2	Descripción del protocolo de red propuesto	23
4.2.1	Algoritmo de selección de ruta de datos mezclados (ADM)	25
4.2.2	Algoritmo agrupamiento y mezcla de datos (AMD)	29
4.3	Algoritmo multicapa propuesto	32
4.4	Diseño de las tramas usadas en el protocolo	33
4.5	Diseño de los tiempos del protocolo	35
4.6	Análisis teórico del protocolo propuesto	39
4.6.1	Retraso de las tramas suponiendo que no hay colisión	39
4.6.2	Análisis probabilístico del retraso de las tramas con la red en reposo	41
4.7	Desarrollo del ambiente de simulación	46
4.7.1	Desarrollo de la Simulación del medio	46
5	Resultados y análisis	51
5.1	Capacidad de autoconfiguración de la red ante fallos en los nodos	53
5.2	Latencia de las tramas	59
5.3	Consumo de potencia	63

6 Conclusiones	67
6.0.1 Trabajo Futuro	68
Bibliografía	69
A Simuladores evaluados	75
B Glosario	81

Índice de figuras

3.1	Vista de diseño de una red en Omnet (Tomado de [24]).	15
3.2	Composición de un nodo en Omnet	16
3.3	Algoritmo FES (tomado de [9]).	17
4.1	Problema de la estación expuesta y la estación oculta, (tomada de[42]).	21
4.2	Ilustración del funcionamiento del protocolo planteado durante la etapa de reconocimiento del problema. (tomada de[42])	22
4.3	Ilustración Conectividad de nodos en una red de sensores	24
4.4	Ilustración del caso en que los nodos tienen mayor alcance de comunicación que los sensores tienen alcance	24
4.5	Diagrama de flujo del algoritmo multicapa propuesto	26
4.6	Ilustración de los estados seguidos por el nodo base	29
4.7	Ilustración de niveles del protocolo de selección de ruta de datos mezclados	30
4.8	Ilustración de distribución de niveles del algoritmo de agrupamiento y mezcla de datos para una red que usa $max_cluster_level = 3$	31
4.9	Gráfico aproximado del consumo de potencia esperado de los nodos con la red en reposo utilizando el protocolo propuesto	36
4.10	En el eje vertical se ve el porcentaje de ahorro en la energía para la red en reposo en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida, mientras que en el eje horizontal se ve que tan grande es el tiempo T con respecto a P	37
4.11	Ilustración del tiempo de retardo mínimo y máximo en unidades de B que tarda una alarma que se da en un nivel dado en llegar a la base asumiendo que no hay contención y que $P = 11B, T = 4P$ y $Max_Cluster_Level = 3$	40
4.12	Función de probabilidad de los tiempos A y B	43
4.13	Función de probabilidad del retraso debido a $A + B$ sin tomar en cuenta el retardo constante debido a $D-H$	43
4.14	Funciones de probabilidad para el retraso para cada una de las posibilidades de selección de C	43
4.15	En (A) se muestra un ejemplo de las funciones de probabilidad independientes del tiempo C y en (B) se muestra la función de probabilidad que se obtiene al sumar el área debajo de las curvas	44
4.16	Fragmento de código de un módulo de definición de red en Mixim (Tomado de [33]).	47

4.17	Fragmento de código del módulo <i>decider</i> en Mixim (Tomado de [33]). . . .	47
4.18	Fragmento de código del módulo de batería en Mixim (Tomado de [33]). . .	49
4.19	Módulos usados en la implementación de los nodos en Omnet++	50
5.1	Configuración física de red para una de las pruebas de autoconfiguración realizadas	53
5.2	topología en cuadrícula para una red de sensores en la que los nodos se encuentran al límite del alcance de radio de sus vecinos	55
5.3	topología en línea para una red de sensores en la que los nodos se encuentran alineados al límite del alcance de radio de sus vecinos]	56
5.4	Retardo de la primera configuración de los nodos de nivel 50 para la topología de malla, en el eje Y se observa el retardo en minutos y en el eje X se observa el número de experimento	57
5.5	Resultado de latencia para las tramas de nivel cincuenta en algunos de los mil experimentos realizados cuando $T = 4P$	59
5.6	distribución de frecuencia del retardo de las alarmas generadas en nodos de niveles 5, 25 y 50 con $T = 4P$	60
5.7	distribución de frecuencia del retardo de las alarmas para saltar de un nodo al siguiente con $T = 4P$	61
5.8	Latencia promedio usando diferentes valores de T para nodos que se en- cuentran en diferentes niveles de <i>ADM</i>	62
5.9	Latencia promedio normalizada en B de las alarmas en llegar a la base para nodos en diferentes niveles de <i>ADM</i> de la red para diferentes niveles de tráfico	63
5.10	Latencia promedio normalizada en B de las alarmas en llegar a la base para nodos en diferentes niveles de <i>ADM</i> de la red para diferentes niveles de tráfico	65
A.1	Ilustración flujo de mensajes en omnet	78
A.2	Configuración de simulación en omnet	79

Índice de tablas

4.1	Descripción de los tipos de alarmas asignadas hasta el momento	34
4.2	Potencia promedio del TR1000 en los diferentes estados de la etapa de comunicación. (Tomada de [36])	36
4.3	Porcentaje de probabilidad de que el retardo de transmisión entre 2 nodos de nivel superior a 1 se encuentre en un intervalo dado	45
4.4	Algunos de los módulos de Mixim usados en la simulación de la red	48
5.1	Resumen de los parámetros de la etapa de radiofrecuencia usados por defecto en las pruebas	52
5.2	Resumen de las parámetros del protocolo usados para las pruebas	52
5.3	Tabla de resumen de la prueba de capacidad de autoconfiguración y efectividad en la entrega de tramas	54
5.4	Datos de tiempo para la obtención de nivel de <i>ADM</i> y <i>AMD</i> en las topologías lineal y en malla para la red en reposo usando valores normalizados en términos de <i>B</i>	56
5.5	Datos de tiempo para la obtención de nivel de <i>ADM</i> y <i>AMD</i> en las topologías lineal y en malla para la red con tráfico usando valores normalizados en términos de <i>B</i>	58
5.6	Datos de tiempo promedio en minutos requerido por nodos de diferentes niveles para la detección de cambios en la red, con la red en reposo y usando topología lineal	58
5.7	Datos de tiempo promedio en minutos requerido por nodos de diferentes niveles para la detección de cambios en la red, con la red con tráfico y usando topología lineal	59
5.8	Datos de tiempo promedio normalizado en <i>B</i> requerido por las tramas enviadas desde diferentes niveles para llegar a la base usando topología lineal, insertando una alarma por minuto en los nodos de nivel múltiplo de cinco con $T = 4P$. Dado que los datos son normalizados no tiene unidades	60
5.9	Datos de tiempo promedio normalizado en <i>B</i> requerido por las tramas enviadas desde diferentes niveles para llegar a la base usando una topología en malla y diferentes frecuencias para la generación de alarmas en 500 nodos aleatorios a la vez. Dado que los datos son normalizados corresponden a unidades de <i>B</i> y no tienen unidades	62

5.10	Tabla de resumen de los valores de potencia promedio de la red en reposo simulados usando diferentes valores de tiempo de hibernación	64
5.11	Datos de potencia promedio en Watts para proporciones de T/P simulados al generar alarmas a diferentes frecuencias	64
5.12	Datos de potencia promedio en Watts para los nodos de diferentes niveles de ADM usando $T = 40$	65

Capítulo 2

Introducción

El creciente desgaste de los recursos naturales que hacen posible la vida en el planeta plantea la necesidad de desarrollar tecnologías que procuren un mayor control sobre los recursos existentes y metodologías de conservación y aprovechamiento ambiental que faciliten el desarrollo sostenible y sostenido de la sociedad.

La degradación de los recursos naturales en el planeta no solamente desemboca en la disminución de las materias primas necesarias en los procesos industriales; sino que además implica el deterioro de la calidad de vida de las personas como consecuencia del agotamiento de los medios necesarios para la subsistencia, tales como el agua potable o la protección atmosférica contra las radiaciones solares.

Incluso en los países que existe preocupación por la conservación de los recursos naturales, como por ejemplo Costa Rica en donde el 25.58% de su territorio se encuentra protegido [44], la tala [6] y la cacería[35] en zonas protegidas por la ley son comunes.

El presente proyecto se circunscribe en el desarrollo de una herramienta tecnológica especializada que permita monitorizar y proteger amplias zonas boscosas usando un mínimo de personal y recursos económicos, mediante el diseño de una red de sensores para la monitorización de variables diversas en el bosque; por ejemplo la detección de sonidos de disparos y motosierras, o la presencia de fuego. Este tipo de información permitiría actuar de forma oportuna ante la presencia de cazadores o leñadores en zonas protegidas, a la vez que facilitaría la respuesta ante incendios forestales. Esto por citar solamente algunas de las posibles aplicaciones de la red.

Para la red de sensores en la que se enmarca la presente investigación, es posible que los nodos se encuentren en zonas de difícil acceso, tales como zonas montañosas o las copas de los árboles, lo cual hace sumamente difícil o imposible el darles mantenimiento a los mismos. Las condiciones anteriormente mencionadas plantean retos especiales al protocolo que se debe emplear en dicha red, ya que el mismo debe ser capaz de operar con un mínimo de consumo energético al tiempo que debe ser confiable, escalable y tolerante a fallos. Entre los principales problemas que se enfrentan en el presente proyecto se encuentran:

- Uso eficiente de los recursos energéticos de los nodos, esto debido a que los módulos de alimentación de energía cuentan con una capacidad limitada de almacenamiento y recarga energética.
- Lograr que la red sea capaz de reconfigurarse por sí misma, ante la falla de alguno de los nodos o adición de nuevos módulos a la red.

En la presente investigación se propone un protocolo de comunicación (algoritmo paralelo orientado a eventos), que cumpla con los requerimientos específicos para la aplicación de la red de sensores de monitorización del bosque anteriormente descrita.

Se propone y evalúa un protocolo que toma en cuenta el estado del arte en protocolos de bajo consumo energético para redes de sensores, al mismo tiempo que toma en cuenta las características específicas de la aplicación anteriormente expuesta con el objetivo de obtener un rendimiento en cuanto al consumo de energía mejor al obtenido usando protocolos genéricos como zigbee. El protocolo propuesto es evaluado con respecto al consumo de potencia usando como referencia la interfaz de red de Texas Instruments CC2420 la cual usa el estandar IEEE802.15.4, mismo estandar usado por zigbee, además se usa wiseroute como protocolo de red. Para hacer la simulación se utiliza Omnet++ y la biblioteca de simulación Mixim. Se obtiene que el protocolo propuesto mejora el consumo de energía de la red en comparación con la red de referencia en reposos en más de un 90% y además se comprueban las características de autoconfiguración, tolerancia a fallas y se provee un modelo para estimar el consumo de potencia promedio de los nodos de la red cuando la misma se encuentra en reposo, que para efectos del presente proyecto se espera que sea la mayor parte del tiempo.

2.1 Objetivos y estructura del trabajo

El objetivo general del trabajo es diseñar un protocolo de comunicación capaz de transmitir alarmas de los nodos de la red a nodos base, utilizando un mínimo de recursos energéticos y económicos.

Como objetivos específicos se han planteado:

- Proponer un protocolo para la para transmisión de alarmas en una red ad-hoc, que tome en cuenta el estado del arte en protocolos de bajo consumo energético, al tiempo que provea un ahorro energético de al menos 20% con respecto de un nodo que implemente el estándar IEEE802.15.4 y que mantenga un tiempo de latencia máximo de 20 minutos para reportar una alarma que se dé en un nodo que necesite 50 saltos para llegar a la base.
- Desarrollar un modelo teórico del protocolo que permita estimar el consumo energético de cada uno de los nodos así como la latencia de las alarmas dependiendo de la cantidad de saltos necesarios para llegar a la base.
- Elegir y configurar un entorno de simulación capaz de evaluar el comportamiento del protocolo ante cambios en la topología de la red, cambios en el ciclo de trabajo

de los nodos y fallos en los nodos que componen la red.

El presente documento esta formado por 4 capítulos a saber:

- **Marco teórico:** en este capítulo se hará una discusión corta acerca de los diferentes protocolos y métodos que se presentan en la literatura y que fueron utilizados como referencia para el desarrollo de la presente investigación.
- **Diseño y modelado teórico del protocolo:** en donde se explica el proceso de modelado y diseño seguido en la definición del protocolo propuesto así como el protocolo en sí.
- **Resultados y análisis:** en donde se detalla el desarrollo del ambiente de simulación y los resultados obtenidos, así como una discusión sobre los resultados.
- **Conclusiones:** en donde se dan las conclusiones de la investigación.

Capítulo 3

Marco teórico

Se dará por sentada en la presente revisión bibliográfica que el lector se encuentra familiarizado con las generalidades del modelo de referencia OSI así como el funcionamiento de protocolos de red básicos como el CSMA. Como una buena referencia en estos temas se recomienda consultar [42].

3.1 Métodos de Ahorro de energía en redes de sensores

Los protocolos diseñados para ser usados en las redes con capacidad de auto configurarse y procesar datos en el sitio, a las que se llamará en adelante “ad_hoc”, para redes convencionales con fuentes de energía estables, se han enfocado en la distribución equitativa del ancho de banda entre los miembros de la red, así como la eficiencia en el envío de grandes cantidades de datos, dejando de lado el consumo de energía, que resulta ser un factor a tomar en cuenta para las redes de sensores [40, 52, 2, 8, 32]. Protocolos tales como el MACAW[5] y el IEEE 802.11[14] reducen el desperdicio de energía causado por la colisión de tramas y se han realizado mejoras sobre estos protocolos (por ejemplo el PAMAS[50]) para evitar la recepción innecesaria de paquetes por nodos que no son el destino final de la trama no obstante su foco sigue siendo la capacidad de transmitir grandes cantidades de datos.

En la investigación disponible como referencia se encuentran técnicas para la reducción del consumo energético en redes de sensores para las siete capas del modelo de referencia OSI. Entre las técnicas encontradas en la en la bibliografía se pueden resaltar:

- **Reducción de los datos:** Se enfoca en la reducción de la cantidad de datos producida, procesada y transmitida por la red, métodos para la compresión y agregación de datos son ejemplos de esta técnica, entre las referencias que utilizan estos métodos se encuentran: [12, 10, 11, 13, 54].

- **Reducción en el costo operativo del protocolo:** El espíritu de esta técnica es la reducción del consumo energético por medio de la reducción o eliminación de operaciones redundantes en el protocolo, normalmente este tipo de optimizaciones se basan en protocolos que se implementan en un solo bloque de software en lugar de usar diferentes capas independientes como sugiere el modelo OSI, otra manera de reducir el costo operativo de la red es identificando los procesos que deben ocurrir de manera periódica en la red y que no son siempre necesarios, por ejemplo los procesos de reconocimiento de vecinos u otros tipos de intercambios de información los cuales si bien es cierto pueden ser necesarios para el funcionamiento correcto del protocolo muchas veces no aportan información nueva. Entre las referencias que utilizan este tipo de técnicas se pueden citar [45, 28, 46, 20].
- **Reducción del tiempo de transmisión:** Esta técnica consiste en la reducción del tiempo en que los nodos mantienen encendida su etapa de transmisión/recepción, entre las referencias que utilizan estas técnicas se pueden citar [23, 19, 50, 45].
- **Control de topología:** Esta técnica utiliza información de la topología de la red para cambiar la potencia de transmisión de los nodos de forma que se crea una topología diferente con un menor alcance de los nodos mientras se mantiene la conectividad, entre las referencias que utilizan esta técnica se pueden mencionar: [40, 15, 53].

Reducción de datos

Para ciertas aplicaciones, si se toma en cuenta la historia de las observaciones pasadas y tomando en cuenta que los nodos de las redes de sensores tienen capacidad de computación, un nodo es capaz de predecir lecturas de los sensores. Las técnicas de predicción de datos están basadas en un modelo, de manera que para ciertos datos los nodos son capaces de usar el modelo para hacer predicciones en lugar de obtenerlos de la red, siempre y cuando dicho modelo asegure un nivel de error aceptable, de manera que se reduce el tráfico y con esto el consumo de energía. Las técnicas de ahorro de consumo energético basadas en modelos de predicción pueden dividirse en dos conglomerados: centralizadas y basadas en agrupamiento. Un ejemplo de una técnica de ahorro de energía basada en predicción centralizada se explica en [12], en donde se muestra que la predicción de datos puede entenderse como *ver una película en las mediciones de los sensores en la red*, y por tanto los conceptos de *MPEG* pueden ser aplicados. Los nodos de la red envían sus datos al sumidero. Este luego calcula el modelo con base en la correlación entre bloques de datos y lo envía de vuelta a los nodos. Futuras lecturas en los sensores serán comparadas contra este modelo y solamente se transmitirán las lecturas que se encuentren fuera del margen de error aceptable. Las técnicas de reducción de datos basadas en agrupamiento toman ventaja de la cercanía espacial entre sensores. Los autores de *ASAP* [10] proponen que los nodos con lecturas similares se agrupen; la cabeza del conglomerado así como los nodos base o sumideros mantienen un modelo de predicción de datos, la cabeza de conglomerado compara los datos de los nodos dentro de su conglomerado contra el modelo y

solamente transmite los datos que se encuentran fuera del margen de error aceptable. El protocolo del *amigo* [11] establece una relación entre los nodos y sus vecinos para explotar la correlación espacio-temporal de los datos. Esto da paso a la creación de *conglomerados de amigos* con un *amigo* líder que se encuentra a cargo de evaluar y procesar los datos. En [13] se propone la creación de conglomerados y cabezas de conglomerados, en donde las cabezas de conglomerado son responsables agrupar y comprimir los datos para luego transmitirlos a la base o sumidero. En [54] los autores proponen *DCTC*, donde cada nodo conoce la distancia al evento detectado, y el nodo más cercano al evento es el que se selecciona como cabeza de conglomerado.

Reducción del costo operativo del protocolo

Los protocolos de comunicación normalmente contienen intercambios de mensajes periódicos. Este intercambio periódico de mensajes es una fuente de desperdicio energético en las redes de sensores. Reducir la frecuencia con que se envían dichos mensajes salva energía y reduce el tráfico de la red, no obstante aumenta la latencia de entrega de los datos. Es necesario determinar cuál es la frecuencia óptima de intercambio de esos mensajes tomando en cuenta esta proporcionalidad inversa entre la latencia y el consumo de energía. Dado que los requerimientos de las redes de sensores inalámbricas cambian de forma dinámica, el periodo de transmisión de dichos datos puede ser también adaptado en forma dinámica, de ahí la idea de intervalos de transmisión adaptativos. El descubrimiento de los vecinos y cálculo de rutas eficientes en términos de consumo energético, son ejemplos de protocolos en los que es necesario intercambiar información periódicamente. En [20] los autores sugieren adaptar la frecuencia de envío de mensajes periódicos dependiendo de la estabilidad de la red; por ejemplo se pueden tener dos frecuencias de envío *hello_min* y *hello_max* para el descubrimiento de vecinos. *hello_max* representa el periodo máximo de envío de mensajes *hello* en una red estable. Este máximo que se usará cuando la red se encuentre en operación estable, no obstante se usa *hello_min* cuando la red se encuentra inestable y se detectan muchos cambios en la topología. En [28] se propone una manera diferente de adaptar el periodo de transmisión de estos mensajes basados en la distancia a la fuente de información, donde el periodo crece con la distancia, por ejemplo la información para los nodos que se encuentran a menos de 3 saltos de la fuente se refrescarán cada periodo, los que se encuentran entre cuatro y seis saltos cada dos periodos, así sucesivamente esto aprovechando que los nodos más alejados a la fuente de información podrían tener necesidades diferentes a los que se encuentran más cerca de la misma. Se han propuesto además protocolos multicapa con el objetivo de reducir el costo operativo de los mismos [46], entre las propuestas se encuentran:

- **Implementación de arriba a abajo:** Donde las capas superiores seleccionan parámetros y estrategias para las capas inferiores. Por ejemplo, la capa de aplicación selecciona los parámetros de la capa *MAC*, mientras que la capa *MAC* selecciona la modulación óptima para la capa física.

- **Implementación de abajo a arriba:** Donde las capas inferiores hacen abstracción de pérdidas y variaciones de ancho de banda para las capas superiores.

Reducción del tiempo de transmisión

Para las redes de sensores, especialmente las que tienen un tráfico bajo de datos, el consumo de potencia debido a la escucha pasiva de los nodos puede ser de la misma magnitud o incluso mayor que la energía consumida en el proceso de transmisión; es por esto que el foco de algunos protocolos de la capa de enlace de datos, específicamente la subcapa *MAC* especialmente diseñados para ser usados en redes de sensores, es el de reducir este tiempo de escucha pasiva, enviando la etapa de comunicación de los nodos a un estado de sueño profundo de manera periódica por el mayor tiempo posible que sea razonable para la aplicación particular que se trata de mejorar. En *S – MAC* [50] cada nodo crea un itinerario de “sueño” que determina en qué momento activa la etapa de escucha del receptor, y cuándo la desactiva para entrar en una etapa de hibernación. Los nodos vecinos no necesariamente tienen que sincronizar sus horarios de sueño; pero deben compartir la información de su itinerario de sueño con los otros nodos durante un proceso de sincronización periódico. Cuando un nodo fuente desea enviar un paquete a un nodo destino, este espera hasta que el destino se “despierte” y envía los paquetes usando *CSMA – CA*. *S – MAC* además introduce un mecanismo de paso de paquetes en el cual los paquetes largos se quiebran en fragmentos que son enviados seguidos de una trama de acuso de recibo. *S – MAC* se evaluó en [50] utilizando una pequeña red de 5 nodos y muestra que existe un compromiso entre el tiempo de latencia de las tramas y el ahorro energético conforme aumenta el periodo de hibernación de los nodos. Protocolos basados en el *S – MAC* ofrecen soluciones para deficiencias y limitaciones del *S – MAC* original, por ejemplo *T – MAC* [45] busca reducir aún más el consumo de energía de la escucha pasiva por medio del cambio dinámico del tiempo activo de los nodos. En lugar de enviar los mensajes usando un tiempo activo predeterminado, como se hace en *S – MAC*, los mensajes son transmitidos en ráfagas al principio del bloque de tiempo. Si no ocurren “eventos de activación” durante un cierto periodo de tiempo, los nodos se ponen en estado de hibernación hasta el siguiente periodo de activación programado. Los “eventos de activación” incluyen la finalización del tiempo de hibernación pre-programado o cualquier actividad de radio, incluyendo datos recibidos o transmitidos. La evaluación utilizada para *T – MAC* en [45] consistió en simulaciones en las que se comparaba su rendimiento en términos de consumo de energía contra *CSMA* clásico y *S – MAC*, los resultados concluyen que bajo carga homogénea *T – MAC* y *S – MAC* logran una reducción similar en el consumo energético, aproximadamente 98% comparado con *CSMA*, sin embargo en un ambiente en el que se tiene carga variable *T – MAC* consume 5 veces más energía que *S – MAC*.

Dado que muchas redes de sensores consisten en árboles de adquisición de datos que tienen su raíz en un solo sumidero, la dirección de los datos que llegan a los nodos es bastante predecible, no así la latencia de las tramas que llegan a los nodos. *DMAC* [19] toma

ventaja de esto al tomar en cuenta la distancia de los nodos a la base para sincronizar el tiempo en el que se despiertan los nodos, *DMAC* reduce el tiempo de latencia de las tramas que son transmitidas desde nodos que necesitan varios saltos para llegar a la base. El modelo para despertar a los nodos consiste de un periodo de recepción y un periodo de transmisión ambos de duración X (donde X es seleccionado para ser la duración de una sola transmisión), seguido por un largo periodo de hibernación. Los nodos inician su periodo de escucha después de un tiempo de espera de $D \times X$ donde D representa la profundidad del nodo en el árbol de adquisición de datos. De esta manera, el periodo de escucha de cada nodo se alinea con el periodo de envío de los nodos de nivel superior, de manera que los nodos simplemente envían durante el periodo de escucha de los nodos de nivel inferior. La evaluación de este protocolo se hace en [19] utilizando un simulador.

En las aplicaciones de redes de sensores es común que los nodos se encuentren constantemente monitorizando el ambiente, y se transmitan datos de manera esporádica a la base, cuando ocurre un evento de interés. *STEM* [37] propone que los nodos se mantengan en estado de hibernación cuando se encuentran monitorizando el ambiente y que se encienda la etapa de comunicación solamente cuando se genera tráfico para transmitir a la base[37]. Se propone que los transceptores se enciendan de manera reactiva y no proactiva como en los protocolos de la capa de control de acceso al medio anteriormente mencionados. Cuando se genera un evento que requiere la transmisión de datos a la base, el sensor que requiere la transmisión del paquete utiliza un canal de transmisión separado del principal y que consume menos energía para despertar al resto de los nodos de la red. Se proponen 2 versiones de *STEM*: *STEM - T* utiliza un “tono” en una frecuencia diferente de la principal para despertar a los nodos vecinos, y *STEM - B* en el que se utiliza el mismo canal de comunicación de manera similar a los protocolos anteriormente mencionados, donde los nodos encienden los canales de comunicación - transmisión periódicamente para ver si existen tramas por transmitir. Para la evaluación del protocolo se usa en [37] un simulador y se comparan las 2 versiones de *STEM* contra un modelo y no contra otros protocolos.

Control de topología

La idea fundamental detrás del control de topología es construir y mantener una topología reducida que ayude a salvar energía en la red, mientras se conserva la conectividad y cobertura de la misma. Esto puede lograrse por medio de la reducción del alcance de la etapa de comunicación de los nodos. En [15] los autores prueban que existe un rango de alcance óptimo que minimiza la disipación de energía y mantiene la conectividad en la topología. En [53] se presenta *READ* “residual energy aware dynamic topology construction algorithm” pensado para redes de sensores heterogéneas donde se toma ventaja de la heterogeneidad de los nodos y se usan los dispositivos con mayor potencia de transmisión para que jueguen un papel más protagonista en la conectividad de la red. En lugar de usar la distancia euclidiana entre dos nodos para definir el costo de la comunicación *READ* introduce un costo para cada par de nodos que toma en cuenta la energía necesaria para

transmitir/recibir la información así como la energía residual en los nodos

3.2 Otros protocolos de control de acceso al medio de bajo consumo energético para redes de sensores

A continuación se hace un breve resumen de algunas de referencias sobresalientes en el campo de los protocolos de bajo consumo de potencia para redes de sensores inalámbricas, haciendo hincapié en las características novedosas que cada uno de los protocolos plantea con respecto de los anteriores.

TRAMA (Traffic-Adaptive Medium Access)

Mientras que los protocolos mencionados anteriormente tratan de reducir el consumo de potencia por medio de la minimización del tiempo de escucha pasiva de la etapa de radio de los nodos, *TRAMA* trata de evitar el gasto innecesario de energía causado por la colisión de paquetes[31]. Los nodos inicialmente intercambian información con sus vecinos usando un *Neighbor Protocol (NP)* de manera que cada nodo tiene conocimiento de todos sus vecinos hasta dos saltos de distancia. Este periodo de intercambio de información con los vecinos es seguido de un periodo de intercambio de información de sincronización en los que los nodos comparten información de sincronización con sus vecinos usando un *Schedule Exchange Protocol (SEP)*. Usando la información de los vecinos adquirida por medio del *NP* y la información de sincronización de los vecinos adquirida usando el *SEP*, los nodos determinan su estado de radio usando lo que el artículo llama *adaptive election algorithm (AEA)*. En *AEA*, cada nodo calcula a priori para sí mismo y para todos los nodos a una distancia de dos saltos la prioridad para tomar un espacio de transmisión. Si un nodo tiene la prioridad más alta para usar el espacio de transmisión y tiene datos para transmitir, entonces gana el derecho a usar el espacio de transmisión. Si alguno de los nodos vecinos tiene la prioridad más alta para ocupar ese espacio de transmisión y el nodo determina que él debe ser el receptor de la trama, usando la información adquirida mediante el *SEP*, entonces el nodo se configura a sí mismo en estado de escucha. En caso contrario el nodo es libre de ponerse en estado de hibernación y ahorrar energía. Dado que es posible que dos nodos en la vecindad de dos saltos de distancia se consideren a ellos mismos ganadores si se encuentran “escondidos” entre ellos, los nodos deben estar pendientes de un ganador alternativo además de un ganador absoluto para cada tiempo de transmisión. Por ejemplo considere un nodo N que determina que el ganador absoluto para un tiempo de transmisión es uno de sus nodos vecinos a dos saltos de distancia $N2 - saltos$. Si un nodo que se encuentra a un salto de distancia de N , y que no conoce a $N2 - saltos$ cree que él es el ganador del espacio de transmisión, y quiere enviar datos hacia N , N debe permanecer despierto aun cuando no considere que $N1 - salto$ haya ganado el derecho de transmitir en ese espacio. Dado que es posible que un nodo gane más espacios de transmisión que los necesarios para vaciar su cola de mensajes por transmitir, algunos espacios de transmisión podrían permanecer vacíos aun cuando pudieron ser usados por

nodos que ganaron menos espacios de transmisión que los necesarios para vaciar su cola de mensajes por transmitir. Para hacer frente a esta situación, el *AEA* asigna prioridades para los espacios no utilizados a los nodos que necesitan más espacios. El rendimiento de *TRAMA* fue evaluado en [31] por medio de simulación y fue comparado contra *SMAC*, *CSMA* y un protocolo llamado *NAMA* el cual se explica con detalle en[4].

IEEE802.15.4 IEEE802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (*low-rate wireless personal area network, LR-WPAN*). Es la base sobre la que se define la especificación de ZigBee [51], cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de la pila de protocolos que el estándar no cubre. Este protocolo utiliza *CSMA – CA*, se cubre con detalle en [22] y será utilizado en el apartado de análisis de resultados como referente para la comparación del protocolo propuesto en el presente trabajo.

3.3 Protocolos de red de bajo consumo energético para redes de sensores

La capa de red para las redes de sensores es sujeto de mejora en el consumo energético debido a que es necesario que las decisiones de selección de ruta de los paquetes tomen en cuenta la cantidad de recursos energéticos disponibles en la red. En caso contrario es posible que algunas de las ramas de comunicación queden sin energía y dejen grupos de nodos aislados (sin acceso a la base) debido a que se utilizó excesivamente dicha rama para la transmisión de paquetes. Es necesario que la cantidad de paquetes enviados en las redes de sensores sea mínima, por lo que la capa de red debe necesitar la menor cantidad posible de transmisión de paquetes para su funcionamiento correcto. En las redes de sensores típicamente los canales de comunicación se dan de los nodos a la base para la transmisión de eventos. Las redes de sensores pueden ser clasificadas como centradas en los datos o centradas en las direcciones, además es posible que los datos sean pre-procesados localmente en lugar de enviar grandes cantidades de datos por la red[27]. Con el objetivo de obtener un protocolo óptimo en el consumo energético es necesario tomar en cuenta las características distintivas de la aplicación o aplicaciones que soportará la red de sensores en específico[3].

Selección de ruta consciente del consumo energético

Este tipo de selección de ruta busca distribuir el consumo energético equitativamente entre los nodos de la red de manera que los nodos no mueran de prematuramente. En [39] se propone que la vida promedio de una red puede ser extendida por medio de la minimización del costo acumulativo C_j de un paquete j enviado del nodo n_1 al nodo n_k

a través de nodos intermedios $n_2, n_3, \text{etc.} \dots$, donde:

$$c_j = \sum_{i=1}^{k-1} f_i(z_i) \quad (3.1)$$

$$f_i(z_i) = \frac{1}{1 - g(z_i)} \quad (3.2)$$

$g(z_i)$ representa el tiempo de vida normalizado correspondiente al nivel de energía z_i del nodo n_i . En [49] se utiliza el modelo propuesto en [39] y se maximiza el tiempo de vida de la red usando un costo de selección de ruta óptimo. En [39] el costo de C_{ij} de una conexión entre un nodo N_i y un nodo N_j se define como:

$$C_{ij} = e_{ij}^{x_1} \underline{E}_i^{-x_2} E_i^{-x_3} \quad (3.3)$$

Donde e_{ij} representa la energía necesaria para transmitir del nodo n_i al nodo n_j , \underline{E}_i representa la energía residual del nodo n_i , y E_i representa la energía inicial en el nodo n_i . En [49] se hace uso de simulaciones para encontrar valores óptimos para x_1, x_2 y x_3 . Basándose en las ideas de este trabajo inicial, se han propuesto otros protocolos conscientes del consumo energético como por ejemplo [38]. En [38] cada nodo tiene un parámetro de costo asociado $Cost(N_i)$ que sirve como una medida de la resistencia del nodo a re-enviar mensajes. Cada vecino de nivel superior N_j del nodo N_i calcula un costo de conexión C_{N_j, N_i} asociado con N_i que depende de $Cost(N_i)$ y de la energía E_{ij} requerida para transmitir a través de esa conexión y de la energía residual R_i del nodo n_i , es decir

$$C_{N_j, N_i} = Cost(N_i) + e_{ij}^\alpha R_i^\beta \quad (3.4)$$

donde α y β son parámetros configurables. Cada nodo N_j construye una tabla de selección de ruta que contiene a sus vecinos con un costo menor de conexión así como el costo C_{N_j, N_i} asociado a cada uno de ellos. El nodo N_j asigna una probabilidad P_{N_j, N_i} a cada uno de sus nodos vecinos en la tabla:

$$P_{N_j, N_i} = \frac{\frac{1}{C_{N_j, N_i}}}{\sum_{k \in FT_j} \frac{1}{C_{N_j, N_i}}} \quad (3.5)$$

de manera que un mensaje recibido por el nodo N_j puede ser re-enviado a sus vecinos con la probabilidad anteriormente mencionada. Antes de reenviar el mensaje N_j el transmisor debe calcular su propio valor de costo que será el promedio ponderado del costo de en la tabla de selección de ruta:

$$Cost(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} C_{N_j, N_i} \quad (3.6)$$

Protocolos de inundación probabilística Estos protocolos se basan en los protocolos de inundación, en los que para enviar un mensaje lo que se hace es simplemente enviarlo a todos los nodos vecinos que lo reenviarán a su vez a todos sus vecinos y así sucesivamente hasta que el mensaje finalmente llegará a su destinatario. Este tipo de protocolos de selección de ruta generan desperdicio de energía porque obligan a todos los nodos a reenviar los mensajes generando duplicación de la información, y colisiones no necesarias. Para mejorar lo anteriormente expuesto, los protocolos de inundación probabilística añaden heurísticas para seleccionar los nodos a los que se le envían los datos, con el riesgo de que el dato finalmente no llegue a su destino como se describe por ejemplo en [47], [30]. Estos protocolos son utilizados en redes de sensores móviles.

WiseRoute WiseRoute[33] es un protocolo de red que crea tablas de selección de ruta desde un nodo central en el que se desee hacer un sumidero de datos. El nodo central envía un paquete de selección de ruta a todos sus vecinos, los cuales lo reenviarán consecutivamente a sus respectivos vecinos, de manera similar a lo descrito para los protocolos de inundación. Una vez que los nodos reciben el paquete de selección de ruta guardan la dirección del nodo o nodos de los que lo recibieron, de manera que todos los nodos conocen uno o varios posibles caminos hacia la base. WiseRoute es el protocolo de red que se usará como referencia para comparar el protocolo propuesto en el presente trabajo.

3.4 Simuladores de redes

Dado que el uso de un simulador para la validación de la propuesta hecha en el presente proyecto es fundamental, a continuación se hace un breve resumen de las características generales de los mismos.

Un simulador es una herramienta que trata de imitar el comportamiento de un sistema. Es posible distinguir entre dos tipos de simuladores: Simuladores basados en tiempo continuo y simuladores de eventos discretos. La mayoría de las herramientas de simulación disponibles son simuladores de eventos discretos. En una simulación de tiempo continuo, el comportamiento del sistema está representado por ecuaciones diferenciales continuas y la simulación consiste en la solución del sistema de ecuaciones. En una simulación de eventos discretos, el comportamiento del sistema se describe como una serie de eventos que deben ser tratados en un punto discreto del tiempo de simulación y que toman una cierta cantidad de tiempo real. Estos eventos son gestionados por un planificador de eventos que permiten, generalmente, simulaciones fuera de línea. No obstante, algunos simuladores integran planificadores de eventos en tiempo real que pueden utilizar como entrada de datos el resultado de una aplicación o de una red real y ofrecer como salida datos que pueden ser enviados a una aplicación especializada o re-inyectados en la red real.

Las herramientas de simulación de eventos discretos generalmente están organizadas de la misma manera. Están compuestas de bibliotecas de modelos y un motor de simulación.

Las bibliotecas de modelos representan los modelos disponibles para describir el sistema que pretende simularse (por ejemplo, para representar elementos de red, enlaces, protocolos o aplicaciones). El motor de simulación ejecuta el escenario de simulación (topología y organización del sistema) y la conducta del escenario (descripción de lo que sucede, donde y cuándo). Un simulador de estas características, generalmente, necesita como entrada de datos el escenario y un comportamiento como de entrada, generando como salida las trazas del resultado de la simulación.

Dependiendo del simulador utilizado, la descripción del escenario y el comportamiento del escenario puede llevarse a cabo desarrollando un archivo de código específico o, de una manera más sencilla, utilizando una interfaz gráfica de usuario. Otros simuladores proponen la generación directa del escenario utilizando algoritmos concretos u otra herramienta auxiliar.

Existe un gran número de simuladores orientados a la simulación de redes. Entre los simuladores que se consideraron para el desarrollo del presente proyecto se encuentran SensorSim[26], J-Sim[29], GloMoSim[43], SENS[41], y NS-2, se selecciona el simulador Omnet++[23] debido a su amplia documentación, utilidades proporcionadas para generación de números aleatorios, generación de estadísticas, topologías entre otras. Además de lo anteriormente mencionado Omnet++ ha sido comparado con NS2 y se han recogido métricas que indican que emplea menos memoria y un menor tiempo de ejecución [48]. En [18] se comparan con detalle ventajas y desventajas de diferentes simuladores. En el apéndice A se muestra un breve resumen de los simuladores evaluados.

Omnet++ es una herramienta de simulación de redes, de eventos discretos, modular y orientada a objetos. Su arquitectura genérica ha permitido que sea utilizada para estudiar problemas de diversos dominios, entre ellos, el modelado de redes de comunicación inalámbrica [25]. La herramienta no es un simulador de algo en particular, sino que provee la infraestructura para escribir simulaciones. Los modelos son ensamblados a partir de componentes reutilizables llamados módulos. Los módulos se comunican entre sí por medio de mensajes a través de compuertas y conexiones, o directamente. Los módulos pueden tener parámetros para personalizar su comportamiento o parametrizar la topología del modelo de red. Al nivel más bajo de la jerarquía de módulos se tienen los módulos simples que encapsulan el comportamiento, se programan en C++ utilizando la biblioteca de simulación. Las simulaciones pueden correrse en varios tipos de ambiente. El ambiente gráfico animado es útil para demostraciones y depuración, el ambiente de línea de comandos es mejor para ejecución por lotes. La herramienta completa es probada en los sistemas operativos más comunes (Linux, MAC OS/X y Windows), y también soporta simulaciones distribuidas. Para este trabajo no se utilizó esta funcionalidad, y las simulaciones corrieron solamente en Linux. Omnet++ es gratis para uso académico y sin fines de lucro [23], y *OMNEST* es su versión comercial.

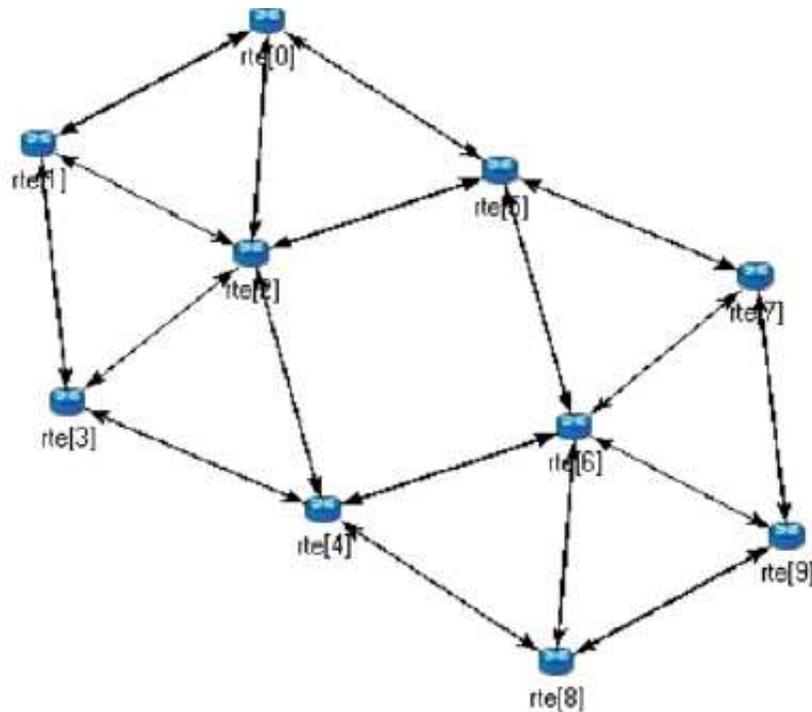


Figura 3.1: Vista de diseño de una red en Omnet (Tomado de [24]).

Conceptos de modelado

Un modelo de Omnet++ consiste de módulos que se comunican a través de mensajes. Los módulos que actúan se llaman módulos simples y se escriben en C++ utilizando la biblioteca de simulación. A su vez, los módulos simples pueden agruparse en módulos compuestos, creando una jerarquía. El modelo completo se llama red y es un modelo compuesto. Los mensajes son enviados por conexiones que comunican los módulos, y pueden contener cualquier información. Típicamente, son enviados por módulos simples a través de compuertas. Las compuertas son las interfaces de entrada y salida de los módulos: los mensajes se envían a través de compuertas de salida y llegan por compuertas de entrada. Una compuerta de entrada y una compuerta de salida se enlazan por una conexión. Se pueden definir tipos de conexión con propiedades específicas (retardo de propagación, tasa de transferencia, tasa de errores de bit *BER*), modelando un canal.

Descripción de red: La estructura del modelo de simulación se describe con el lenguaje *NED Network Description*. Con este lenguaje se declaran módulos simples y módulos compuestos. Un módulo compuesto puede ser etiquetado como "red", es decir, como modelo de simulación auto-contenido. Por ejemplo, se puede definir una red compuesta por nodos intermedios, donde en cada nodo hay una aplicación corriendo, que genera paquetes de datos transmitidos a través de un canal (ver figura 3.2, obtenida del manual de la herramienta[24]).

El nodo es un módulo compuesto por los submódulos que modelan la aplicación y el encaminamiento (capa física, acceso al medio y capa de red). La figura es la vista de

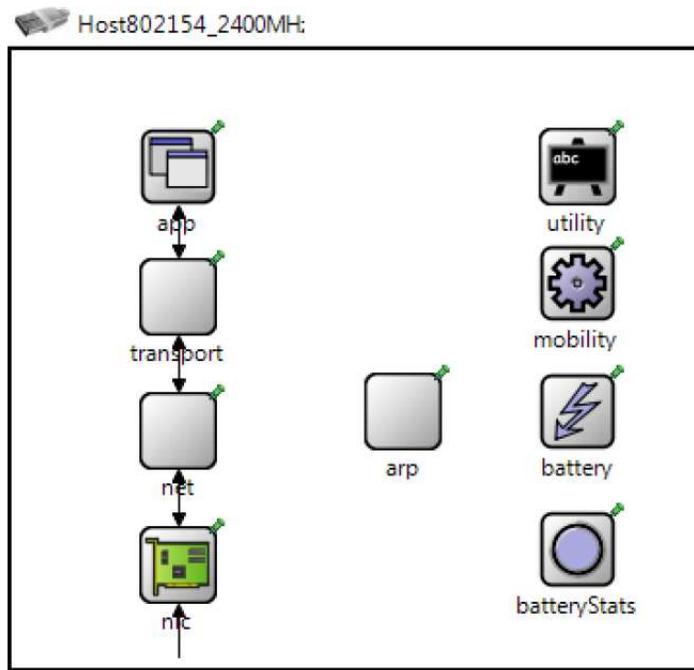


Figura 3.2: Composición de un nodo en Omnet .

diseño de un nodo MiXiM.

Conceptos de simulación

Como ya se ha dicho, los módulos que actúan se llaman módulos simples y se escriben en C++ utilizando la biblioteca de simulación. Los siguientes conceptos de simulación de eventos discretos permiten entender cómo se diseñan los módulos simples.

Simulación de eventos discretos: Un sistema de eventos discretos es aquel en el que los cambios de estados suceden en tiempo discreto, y el evento emplea cero cantidad de tiempo en suceder. Se asume que nada interesante sucede entre dos eventos consecutivos, es decir, no se produce ningún cambio de estado. Este tipo de sistema puede ser modelado usando simulación de eventos discretos. El tiempo en el que el evento ocurre se llama marca de tiempo y en Omnet++ se llama tiempo de llegada (ya que en la biblioteca de clases la palabra marca de tiempo está reservada para un atributo asignable en la clase evento). El tiempo en el modelo se llama tiempo de simulación, tiempo de modelo o tiempo virtual en oposición al tiempo real, que se refiere al tiempo durante el cual el programa ha estado corriendo, o el tiempo de CPU que se refiere a cuánto tiempo de CPU ha consumido.

Eventos: En Omnet++ los eventos son representados por mensajes, es decir, por una instancia de la clase `cMessage` o una subclase de esta. Los mensajes son enviados desde un módulo a otro, lo que significa que el lugar donde el evento ocurre es en el módulo destino, y el tiempo de modelo en el que ocurre es el tiempo de llegada del mensaje. Los eventos se consumen en el orden que da el tiempo de llegada. Si el tiempo de llegada es

FES: Ciclo de evento

```
1 inicializar (construye el modelo e inserta eventos iniciales )
2     while FES no vacía Y simulación no completa do
3     obtener primer evento de FES
4     t := marca de tiempo de este evento
5     procesar evento (inserta nuevos eventos o borra eventos de FES)
6 end
7 terminar simulación (escribir resultados estadísticos, etc.)
```

Figura 3.3: Algoritmo FES (tomado de [9]).

el mismo, el orden lo da la prioridad del mensaje. Si la prioridad es la misma, el orden es aleatorio.

El ciclo de eventos: La simulación de eventos discretos mantiene una lista de eventos futuros en una estructura comúnmente llamada *FES* [9] (*Future Event Set*). En Omnet++ se implementa como un montículo binario. El simulador trabaja de acuerdo al pseudocódigo de la figura 3.3 [9].

El paso de inicialización construye las estructuras de datos que representan el modelo a simular, ejecuta el código de inicialización que se ha redefinido, e inserta eventos iniciales en la lista de eventos futuros. El ciclo subsiguiente consume eventos de la lista y los procesa. Los eventos se procesan en orden estricto dado por la marca de tiempo, para mantener la causalidad. Procesar eventos involucra la invocación de código provisto por el módulo creado. La simulación se detiene cuando no hay más eventos o cuando se alcanza el límite de tiempo de modelo o CPU. Antes de terminar la ejecución, generalmente se graban estadísticas en archivos de salida.

Capítulo 4

Diseño del protocolo

Con el objetivo de reducir el consumo energético y cumplir con los objetivos de la aplicación, el protocolo propuesto debería tener las siguientes características:

- **Reducir las colisiones y minimizar del tráfico en la red:** Dado que se asume que todos los nodos comparten el mismo canal de comunicación es necesario que las colisiones sean reducidas al máximo para evitar la retransmisión de datos, además se desea que el protocolo use la menor cantidad posible de tramas de control para reducir aún más el consumo energético y la probabilidad de colisión de las tramas. Para esto se propone la creación de conglomerados que reciban la información de los nodos vecinos, y que generen tramas que compriman y que mezclen la información, de forma que se reduzca la cantidad de tramas circulando en la red y la probabilidad de retransmisión debida a colisiones. Al usar un protocolo multicapa se propone el uso de las mismas tramas de control usadas para la administración del canal para la administración de la capa de red.
- **Debe procurar el éxito del envío de los mensajes:** El protocolo propuesto debe contener mecanismos que garanticen que el mensaje sea recibido de manera exitosa en el siguiente nodo. Para esto se propone el uso de tramas de reconocimiento de recibido, a las cuales se llamará en adelante *ACK*, y contendrán un chequeo de suma *CHECK_SUM* o algún otro método para verificar que la trama fue recibida correctamente
- **Debe apagar la etapa de radio por la mayor cantidad de tiempo posible:** Especialmente en una aplicación como la que se plantea en el presente proyecto, en la que se asume que la mayoría del tiempo la red se va a encontrar en reposo y monitorizando el ambiente, es necesario que los nodos apaguen la etapa de radio si no es necesaria, para ahorrar energía. Para esto se propone el uso de ciclos periódicos de hibernación en todos los nodos
- **Debe tener una latencia aceptable:** Dado que se busca reducir el consumo de energía en los nodos, esto normalmente tiene un impacto directo en la latencia

de las tramas y es importante acortar la latencia para que quede dentro de un margen aceptable para la aplicación planteada. En este sentido se propone modelar el retardo de las tramas con respecto al tiempo de hibernación de los nodos, para que sea posible configurar la red de manera que se cumpla con los requerimientos de la aplicación a la que se desee dar servicio.

- **Selección de la ruta más corta posible:** Es necesario que los paquetes transiten por la ruta más corta posible, de manera que se reduzcan los saltos innecesarios, que no solamente van a hacer más largo el periodo de latencia de la trama, sino que además van a gastar energía de manera innecesaria. En este sentido se propone un protocolo en el que los nodos sepan su distancia en cantidad de saltos a la base y sean capaces de escoger la ruta más corta para enviar sus datos.
- **Auto-reconfigurable:** El protocolo de red debe ser capaz de reconfigurarse conforme algunos de los miembros de la red se queden sin energía de forma que sea capaz de mantener la funcionalidad de la red por más tiempo. Para cumplir con este requerimiento cada nodo monitoriza el ambiente de forma periódica para detectar cualquier cambio en la topología de la red.

4.1 Descripción del protocolo de control de acceso al medio propuesto

Durante el proceso de investigación previa se decidió evitar el uso de protocolos llamados “libres de colisiones”, en los que se le asigna a cada uno de los nodos en un dominio de colisión un canal, o un espacio de tiempo en el que puede transmitir sin competidores; un ejemplo de estos protocolos son los que usan *TDM* para la administración del canal. Estos protocolos suponen un mayor consumo de energía en los nodos encargados de regular el uso del canal, o el espacio de tiempo que corresponde a cada uno de los nodos para transmitir, esto debido a que estos nodos deben sincronizar la transmisión de los nodos que se encuentran alrededor. Si bien es cierto que este tipo de protocolos reducen el desperdicio de energía por retransmisiones causadas por colisiones en los nodos que no están administrando el canal, también es cierto que la aplicación específica para la cual se está diseñando es una aplicación de bajo tráfico, en la que se espera que se transmitan tramas de forma esporádica, por lo que se considera de mayor ganancia mantener un protocolo donde todos los nodos sean capaces de mantener periodos de sueño tan prologados como sea posible y no transmitir tramas a menos que sea estrictamente necesario.

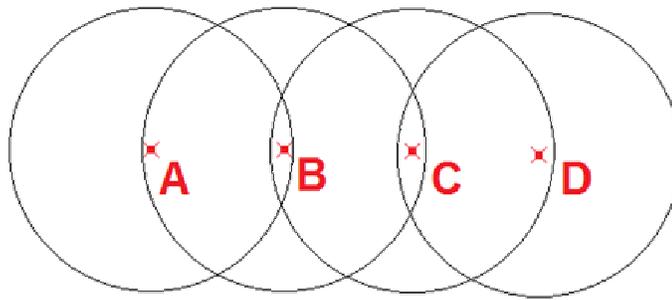


Figura 4.1: Problema de la estación expuesta y la estación oculta, (tomada de[42]).

Un enfoque ingenuo para un protocolo de control de acceso al medio en una red inalámbrica de sensores podría ser intentar algo como el CSMA tradicional en el que se escucha si hay otras transmisiones en el canal y solo se transmite si nadie más lo está haciendo. Este protocolo no es realmente adecuado porque lo que importa en las comunicaciones inalámbricas es la interferencia en el receptor, no en el emisor. Para entender la naturaleza de este problema, en la figura 4.1 se ilustran cuatro estaciones inalámbricas.

El alcance de radio es tal que A y B están en el mismo alcance y potencialmente pueden interferir entre sí. C también podría interferir tanto con B como con D , pero no con A . Cuando A está transmitiendo hacia B . Si C detecta el medio no podría escuchar a A porque esta fuera de su alcance y por tanto deducirá falsamente que puede transmitir a B . Si C comienza a transmitir, interferirá en B eliminando la trama de A . El problema de que una estación no pueda detectar a un competidor potencial por el medio, puesto que dicho competidor esta demasiado lejos se denomina problema de estación oculta.

Considérese ahora la situación opuesta: B transmitiendo a A . Si C detecta el medio, escuchará una transmisión y concluirá equivocadamente que no puede enviar a D , cuando de hecho tal transmisión causaría una mala recepción solo en la zona entre B y C , en la que no esta localizado ninguno de los receptores pretendidos. Esta situación se denomina problema de estación expuesta.

El problema es que antes de comenzar una transmisión, una estación realmente necesita saber si hay actividad o no alrededor del receptor. El CSMA simplemente le indica si hay o no actividad alrededor de la estación que esta detectando la portadora. Con un cable, todas las señales se propagan a todas las estaciones de manera que solamente puede llevarse a cabo una transmisión en un momento dado en un lugar del sistema. En un sistema basado en ondas de radio de corto alcance, pueden ocurrir transmisiones simultáneas si las ondas tienen destinos diferentes y estos están fuera de alcance entre sí.

Tomando en cuenta lo anterior se plantea un protocolo cuya estrategia sea que el emisor estimule al receptor a enviar una trama corta de manera que las estaciones cercanas puedan detectar esta transmisión y eviten ellas mismas hacerlo durante la siguiente trama de datos. Este protocolo es el que se conoce como MACAW (Acceso múltiple con prevención de colisiones inalámbrico), y fue desarrollado inicialmente por Karn, Bharghavan, y Cols entre 1990 y 1994[42].

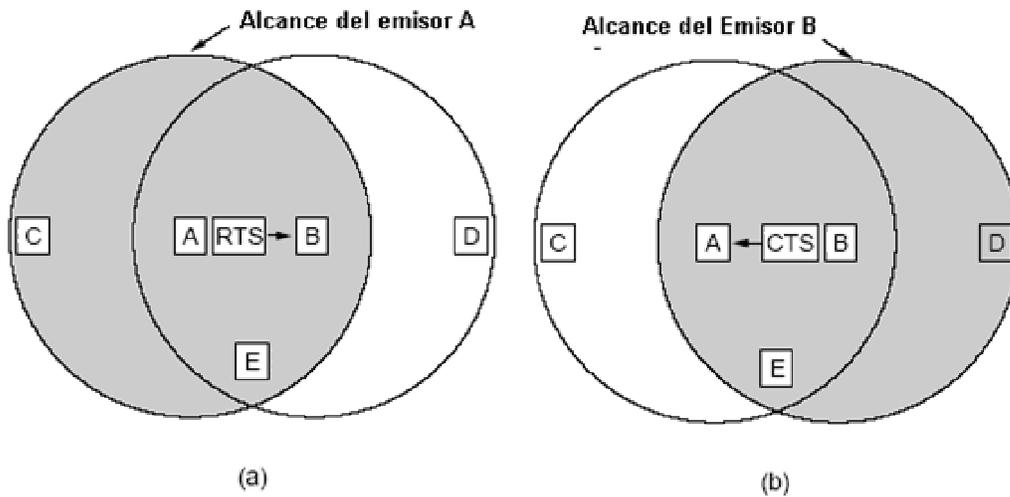


Figura 4.2: Ilustración del funcionamiento del protocolo planteado durante la etapa de reconocimiento del problema. (tomada de [42])

En la figura 4.2 se ilustra el concepto. Considere la manera en que A envía una trama a B . A comienza por enviar una trama RTS (solicitud de envío) a B , como se muestra en la figura 4.2 (a), esta trama contiene la longitud de la trama que seguirá posteriormente; se puede hacer la trama más corta si se hace que esta longitud tenga un valor fijo. Después B contesta con una trama CTS (Libre para el envío) como se muestra en la figura 4.2 (b) la trama CTS contiene la longitud de los datos copiada de la trama RTS . Una vez que sucede la recepción de la trama CTS , A comienza a transmitir. Cualquier estación que escuche el RTS evidentemente está bastante cerca de A y debe permanecer en silencio el tiempo suficiente para que el CTS se transmita de regreso a A sin conflicto. Cualquier estación que escuche el CTS está bastante cerca de B y debe permanecer en silencio durante la siguiente transmisión de datos, cuya longitud puede determinar examinando la trama CTS . En la figura 4.2, C está en el alcance de A , pero no en el de B . Por lo tanto escucha el RTS de A pero no el CTS de B . Mientras no interfiera con el CTS está libre para transmitir mientras se está enviando la trama de datos. En contraste, D está en el alcance de B pero no de A . No escucha el RTS pero si el CTS . Al escuchar el CTS se le indica que está cerca de una estación que esta a punto de recibir una trama, por lo que difiere el envío de cualquier cosa hasta el momento en que espera la terminación de esta trama. La estación E escucha ambos mensajes de control y, al igual que D , debe permanecer en silencio hasta que haya completado la trama de datos.

A pesar de estas precauciones, aún pueden ocurrir colisiones. Por ejemplo, B y C pueden enviar tramas RTS a A . Al mismo tiempo estas chocarán y se perderán. En el caso de una colisión, un emisor que no escucha un CTS en el intervalo de tiempo esperado espera un tiempo aleatorio y reintenta.

Se plantea el uso de una trama ACK (reconocimiento) tras cada trama de datos transmitida exitosamente que provea al transmisor de una confirmación de que la trama se recibió de manera exitosa y no es necesario enviar nuevamente la trama.

El protocolo anteriormente expuesto asume que las estaciones inalámbricas se encuentran encendidas todo el tiempo de manera que las estaciones cuando no se encuentran transmitiendo se encuentran escuchando el medio. En el protocolo que se propone en el presente proyecto los nodos no se encuentran siempre encendidos por lo que antes de iniciar el proceso de transmisión de tramas es necesario que los nodos se escuchen por un tiempo prudencial para asegurar que no se perdieron de escuchar una trama de *CTS* o una de *RTS*.

Dado que se propone apagar los nodos periódicamente y no se quieren usar mecanismos para la sincronización de los periodos de hibernación de los nodos debido a que esto supondría un mayor gasto energético, es necesario utilizar algún tipo de señalización para avisar a los nodos vecinos cuando uno de los nodos se encuentra listo para retransmitir tramas provenientes de nodos que se encuentra más alejados de la base. Con este Objetivo se propone el uso de una trama a la que se llamará *PT* que son las siglas de "petición de transmisión", esta trama se transmitirá después de que el nodo se despierte, una vez terminado el tiempo prudencial anteriormente mencionado para escuchar las tramas de *RTS* y *CTS*. El tiempo que se debe escuchar después de que uno nodo despierta y antes de transmitir la trama de *PT* no debe ser menor que el tiempo necesario para transmitir la trama más larga posible en el protocolo, a este tiempo se le llamará en adelante *B* y será un parámetro fundamental para la configuración de la red.

En el presente proyecto se decide diseñar un protocolo en el que la implementación se hace en un solo bloque de lógico de software, en lugar de que el protocolo sea multi-capa, o sea uno en el que las funciones del modelo de referencia *OSI* se implementan usando capas independientes de software, debido a que tradicionalmente el uso de las capas si bien es cierto facilita el proceso de codificación y desarrollo de las redes, también añade un esfuerzo extra en términos de tareas duplicadas y procesamiento de datos que conlleva un mayor gasto energético. En el presente documento se explicarán los protocolos/algoritmos propuestos en el contexto de las capas del modelo de referencia *OSI*, no obstante en la implementación no se hará en capas, sino que se hará en un solo bloque para evitar la duplicación de tareas y paquetes.

4.2 Descripción del protocolo de red propuesto

La conexión de los nodos en una red de sensores inalámbrica es aproximadamente una estrella extendida, en la que la conexión se encuentra condicionada al alcance de la etapa de radiofrecuencia de los nodos; como se muestra en la figura 4.3

Partiendo de esto se propone una red organizada jerárquicamente en niveles, en la que se pueden definir como nodos de nivel 1 todos los nodos que se encuentren en el alcance de la estación base (zona gris), los nodos de nivel 2 serán los que se encuentren en el alcance de algún nodo de nivel uno y no de uno de nivel cero (zona amarilla) y así sucesivamente. Es necesario que los mensajes transmitidos por un nodo de nivel *N* pasen en el mejor de

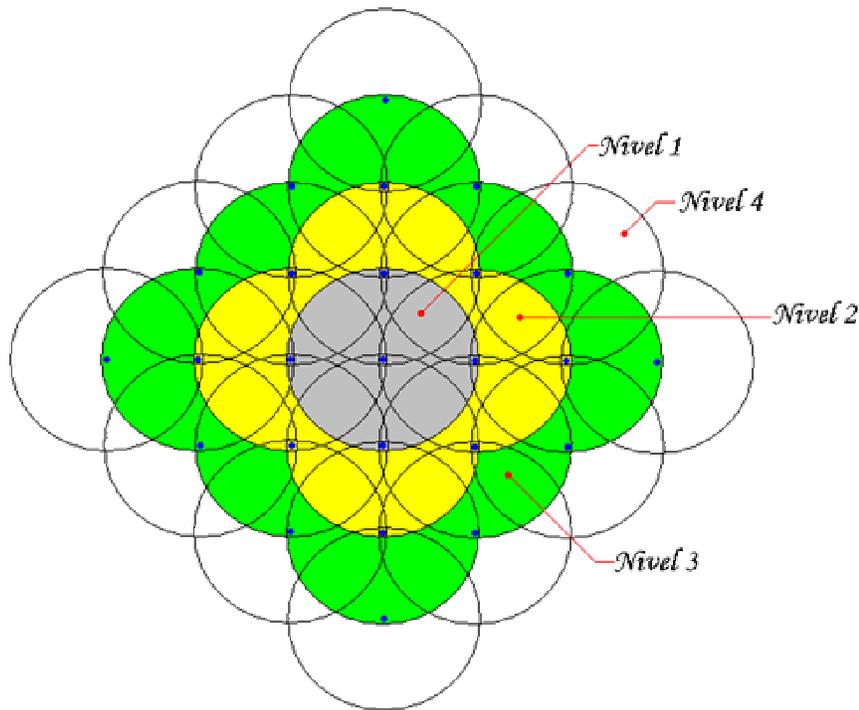


Figura 4.3: Ilustración Conectividad de nodos en una red de sensores

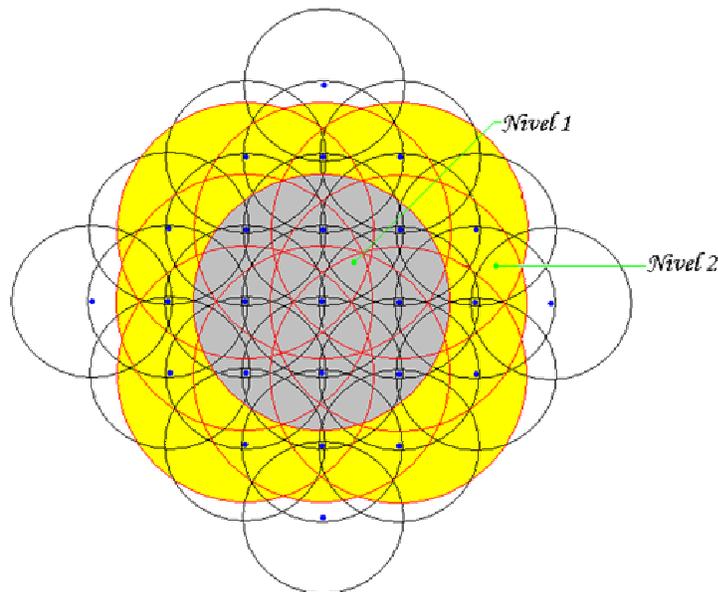


Figura 4.4: Ilustración del caso en que los nodos tienen mayor alcance de comunicación que los sensores tienen alcance

los casos por $N - 1$ saltos antes de llegar a un sumidero. Además si se pretende cubrir el 100% del área sin dejar regiones sin cubrir en medio de los sensores, la posición geográfica de cada uno de los nodos no dependerá del alcance del módulo de radiofrecuencia del nodo solamente, sino del alcance de los sensores en los nodos.

La principal consecuencia de esto es que pueden haber N nodos de nivel k dependiendo de la ubicación que se le asigne a cada uno de los nodos, y pueden haber N nodos en un

dominio de colisión dado. Por ejemplo en la figura 4.4 los nodos se colocan de acuerdo con el radio de alcance del sensor que abarca un área menor (este radio se representa por medio de los círculos negros) mientras que los niveles de la red siguen siendo definidos en función del alcance del módulo de comunicación (los cuales se representan por medio de los círculos rojos).

En el presente documento el uso de ejemplos e ilustraciones en las que se muestran nodos con conexiones en línea vista y donde se tiene un 100% de cobertura se hace porque este es el peor caso para la generación de dominios de colisión, no obstante el protocolo propuesto a continuación es capaz de soportar diferentes configuraciones de conectividad y bloqueos geográficos entre los nodos.

El algoritmo propuesto se puede dividir en dos secciones concurrentes. Para simplificar su comprensión se explicarán estas dos partes por separado y luego se unirán para formar el algoritmo completo.

4.2.1 Algoritmo de selección de ruta de datos mezclados (ADM)

Como su nombre lo indica este es el algoritmo con el que se enviarán los datos que ya han sido mezclados hacia los sumideros de la red.

Se propone que cada uno de los nodos pase a través de una serie de estados que contemplan un estado de hibernación para reducir el consumo de energía en el módulo de comunicación; Además se procura que el protocolo tenga la capacidad de configurar de manera automática la arquitectura de la red evitando de esta manera la intervención externa en el proceso de instalación y/o ampliación del sistema.

El protocolo propuesto se describe de forma gráfica en la figura 4.5 , en la que se esboza la manera en la que cada uno de los nodos pasa de un estado a otro, añadiendo además el protocolo de control de acceso al medio anteriormente propuesto en el mismo diagrama, de forma que las tramas de *RTS*, *CTS* y *ACK* son ya tomadas en cuenta en los estados propuestos. A continuación se explica la función de cada uno de los estados del protocolo desde el punto de vista del algoritmo de selección de ruta de datos mezclados *ADM*.

Descubrimiento de nivel de pertenencia: En esta etapa el nodo descubre su nivel de pertenencia en la red; esto lo hace basándose en la definición de nivel que se proporcionó anteriormente: Se definen como nodos de nivel uno todos los nodos que se encuentren en el alcance de la estación base, los nodos de nivel dos serán los que se encuentren en el alcance de algún nodo de nivel uno y así sucesivamente. De esta definición se puede entender claramente que la pertenencia a un nivel u otro se encuentra en función del radio de alcance del transmisor del nodo de nivel inmediatamente inferior y la posición geográfica de los nodos.

En este estado se escucha el medio durante un tiempo suficiente para detectar las tramas *PT* de todos los nodos cercanos y comparar su nivel en la red. Una vez que se tienen los niveles de todos los nodos cercanos se selecciona el menor nivel escuchado y se sabe que

el nodo que escucha se encontrará en el nivel menor escuchado más uno. Al finalizar este proceso se procede a enviar una trama de alarma especial que indica que el nodo entró en operación. En este caso se plantea el uso de la alarma cero. En caso de que el nodo escuche por el tiempo predeterminado para este periodo sin oír una sola trama de *PT*, el nodo pasará al estado de hibernación para ahorrar energía para pasar nuevamente al estado de descubrimiento de nivel de pertenencia, repitiendo el ciclo hasta que se tenga un nivel de red en el nodo.

Hibernación: En este estado no se reciben ni se transmiten datos; la etapa de radiofrecuencia se apaga completamente para conservar la energía. Este estado finaliza cuando termina el tiempo destinado al mismo. Una vez que finaliza se revisa si existen alarmas por enviar. Si hay alarmas se pasa al primer estado de espera y en caso contrario a la primera fase de petición de trama. Además, tiene un contador que lleva el control de cuántas veces se ha pasado por este estado; Si este contador detecta que el protocolo ha pasado más de X veces por este estado entonces el nodo regresa al estado de descubrimiento del nivel de pertenencia para verificar la arquitectura de la red y redefinir su nivel de pertenencia en la red. X será un parámetro de la red y se definirá dependiendo de las necesidades de la misma.

Primera fase de petición de trama: En este estado el nodo escucha por un periodo de tiempo $2B$ para evitar colisiones con tramas que se encuentren en proceso de transmisión por medio de la escucha del canal en busca de tramas *RTS* o *CTS*. En caso de no escuchar tramas de *CTS* o *RTS* envía una trama *PT* preguntado a los nodos de nivel superior si tienen una trama para retransmitir en caso de no recibir respuesta el nodo vuelve al estado de hibernación. En caso de que algún nodo escuche una trama *PT* se comportará de la misma manera que si hubiese escuchado una trama de *RTS* si se encuentra en la fase de escucha de la primera fase de petición de trama.

Segunda fase de petición de trama: En caso de que alguna de las tramas *PT* descritas en el estado anterior sea contestada con una trama *RTS*, el nodo pasa a la segunda fase de petición de trama; en este estado el nodo envía una trama *CTS* para informar al primer nodo que envió el *RTS* que tiene el canal asignado para la transmisión de la trama, los nodos que escuchen el *CTS* se mantienen en silencio mientras se transmite el mensaje con el objetivo de evitar que las tramas colisionen, de la manera que se explicó anteriormente en la descripción del protocolo de control de acceso al medio.

Escucha: Una vez que el nodo envía el *CTS*, espera para recibir el mensaje del nodo que solicitó el canal; si el tiempo de espera se acaba, el nodo asume que el transmisor tuvo un fallo y retorna al estado de hibernación. Si se recibe la trama a retransmitir, el nodo pasa al estado de verificación.

Verificación: En este estado se envía una trama de *ACK* y se espera un tiempo para determinar que la trama se haya recibido correctamente; si el nodo emisor de la trama no corrige el mensaje enviado se asume que el mismo se recibió correctamente y se pasa al primer estado de espera, en caso de que la trama sea corregida se guarda la nueva información de la alarma y se inicia nuevamente el estado. La trama de *ACK* contiene

un código de *CHECK_SUM* que permite verificar que la trama recibida en el receptor es la correcta. Si la alarma que se recibe es correcta, se guarda y se pasa nuevamente al estado de primera fase de petición de trama para ver si hay algún otro nodo que necesite reenviar información.

Primer estado de espera: En este estado el nodo tiene alarmas por transmitir a al base y espera por una trama *PT* de algún nodo de nivel inferior; en el momento en que dicha trama es escuchada el nodo pasa al segundo estado de espera. Si el nodo escucha una trama *PT* de un nodo de nivel inferior al inmediatamente inferior al propio, cambia su identificador de nivel para ajustarse a la nueva configuración de la red. En caso de no recibir una trama de *PT* válida después de un tiempo el nodo asume que hubo un cambio en la red y vuelve al estado de descubrimiento de nivel sin borrar la memoria de alarma. Este estado tiene una doble función ya que además de tener un papel en el proceso de transmisión de los datos también tiene el propósito de auto-configurar la red en caso de que ocurra un cambio en la topología que la rodea. En caso de escuchar un *CTS* o un *RTS* el nodo esperará por el tiempo requerido para que la transmisión en progreso sea completada e ignorará cualquier *PT* que escuche en este tiempo.

Segundo estado de espera: Después de ingresar al segundo estado de espera el nodo aguarda un tiempo aleatorio corto, esto con el objetivo de evitar colisiones con algún otro nodo que desee contestar la transmisión *PT*, y envía una trama de *RTS* para luego pasar al tercer estado de espera.

En caso de escuchar un *CTS* proveniente del nodo del que escuchó la trama de *PT* en el periodo de espera, se incrementa un contador y el nodo regresa al primer estado de espera.

Si el contador indica que el nodo ha pasado por este estado 3 veces entonces en lugar de pasar al primer estado de espera el nodo asume que hay muchos nodos que desean transmitir sus tramas y pasa al estado de hibernación sin borrar la alarma.

Tercer estado de espera: En este estado el nodo espera por un *CTS* dirigido hacia él mismo. En caso de ser recibido el nodo pasa al cuarto estado de espera. En caso de que no reciba la trama *CTS* el nodo incrementa un contador que indica la cantidad de veces que el nodo ha pasado por este estado y pasa al primer estado de espera. Si el contador indica que el nodo ha pasado por este estado 3 veces entonces en lugar de pasar al primer estado de espera el nodo asume que hay muchos nodos encendidos y pasa al de hibernación sin borrar la alarma.

Cuarto estado de espera: En este estado el nodo envía la trama de alarma y espera por una trama de *ACK* (de reconocimiento de envío) que indica que la información llegó correctamente al nodo receptor; en caso de no recibirse después de un tiempo de espera *B*, el nodo pasa al estado de hibernación sin limpiar la memoria de la alarma. En caso de que el nodo reciba la trama de *ACK* se verifica que la información recibida sea correcta. Si no es así se retransmite la trama y se espera nuevamente el *ACK* (o sea se vuelve a iniciar el cuarto estado de espera). En caso de que la información sea recibida correctamente se

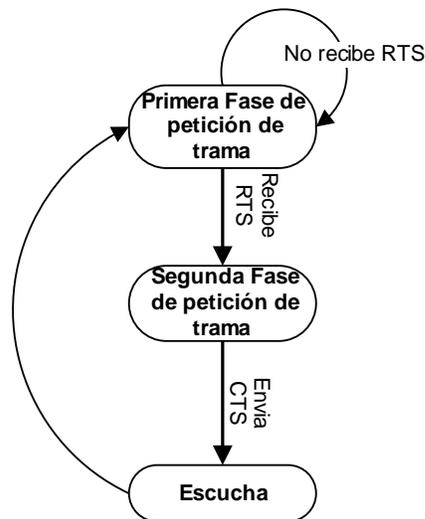


Figura 4.6: Ilustración de los estados seguidos por el nodo base

limpia la memoria de la alarma y se pasa al estado de hibernación.

Nodo base: La estación base solamente escucha para verificar las transmisiones de los nodos de nivel uno y no detecta eventos.

El diagrama de estados que se sigue en este nodo se muestra en la figura 4.6.

4.2.2 Algoritmo agrupamiento y mezcla de datos (AMD)

La naturaleza de la aplicación con la que se está trabajando, hace que sea posible que más de un nodo de la red detecte un evento al mismo tiempo, de manera que ante por ejemplo el disparo de un arma de fuego, sean muchos los nodos que deseen informar de la alarma en la misma área geográfica.

Con el objetivo de minimizar la cantidad de datos transitando en la red y por ende reducir el consumo energético, se propone que los nodos, antes de usar el algoritmo de selección de ruta de datos mezclados expuesto en el apartado anterior para dirigir la información hacia los sumideros finales, compriman y mezclen la información que se enviara a los sumideros en conglomerados locales. Para lograr lo anterior es necesario definir:

- Un algoritmo para formar los conglomerados locales y la cabeza del conglomerado.
- Un mecanismo para la compresión de los datos recolectados en las cabezas de conglomerados.

Además es necesario que el algoritmo propuesto sea compatible con el protocolo de selección de ruta de datos mezclados con el objetivo de reducir o eliminar el uso de tramas de control adicionales a las ya utilizadas en el protocolo de selección de ruta de datos mezclados. Basados en lo anterior se propone que el algoritmo de agrupamiento y mezcla de datos funcione igual que el algoritmo de selección de ruta de datos mezclados con la

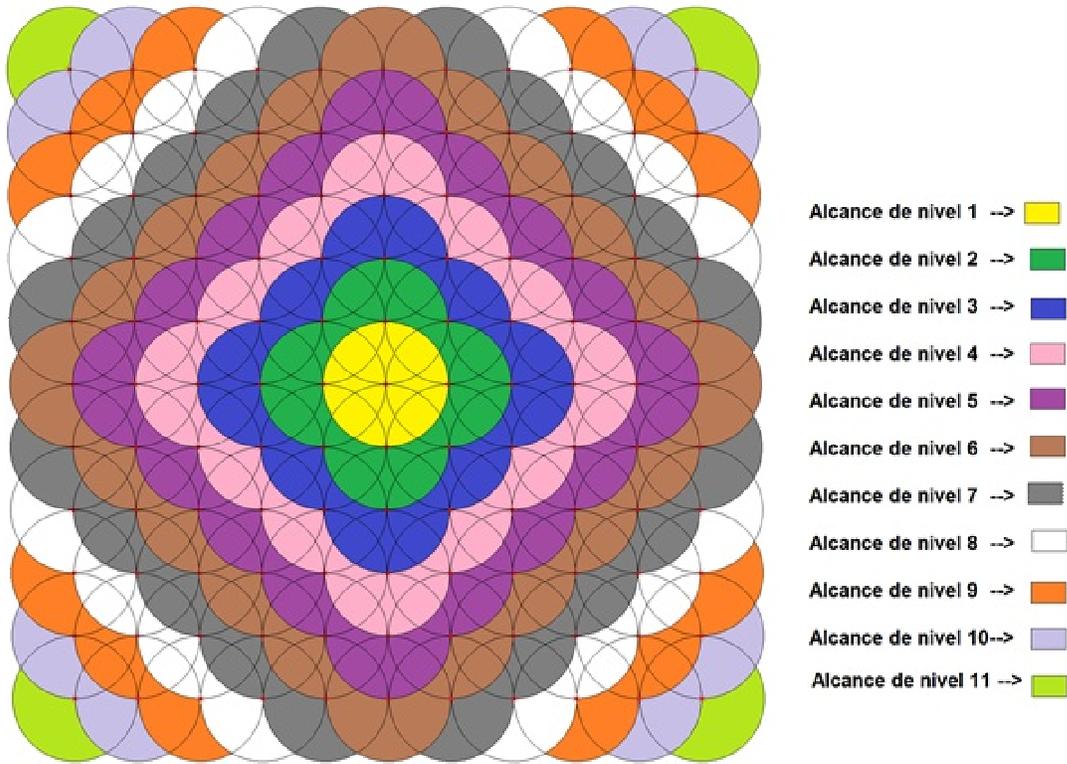


Figura 4.7: lustración de niveles del protocolo de selección de ruta de datos mezclados

diferencia que cada una de las cabezas de conglomerado funcionará como un sumidero local en el que se acopiarán las alarmas, y se enviarán los mensajes usando algoritmo de selección de ruta de datos mezclados. No es el objetivo de la presente investigación el ahondar en los algoritmos disponibles para compresión de datos en las cabezas de conglomerado, sino el proveer el protocolo de comunicación que permita tener los grupos de datos en las cabezas de conglomerado, de manera que se deja a los futuros implementadores del protocolo el seleccionar el mejor algoritmo de compresión que se ajuste a las necesidades de la aplicación particular.

Algoritmo para formar los conglomerados locales y seleccionar la cabeza del conglomerado:

Asuma que se tiene una red en la que los nodos se encuentran posicionados como se muestra en la figura 4.7. Lo primero que debe ocurrir en esta red es que cada nodo debe descubrir su nivel en la red.

Una vez que cada nodo conoce su nivel de pertenencia en la red para el algoritmo de selección de ruta de datos mezclados es necesario seleccionar las cabezas de conglomerados, e iniciar el proceso de descubrimiento de nivel de pertenencia en el algoritmo de agrupamiento y mezcla de datos. Para esto es necesario seleccionar un nivel máximo de saltos permitidos en los conglomerados, al cual se le llamará: "*Max_cluster_Level*" y de aquí en adelante se considerará como parámetro de la red.

Dependiendo de la cantidad de saltos máxima seleccionada se usarán los nodos de "nivel

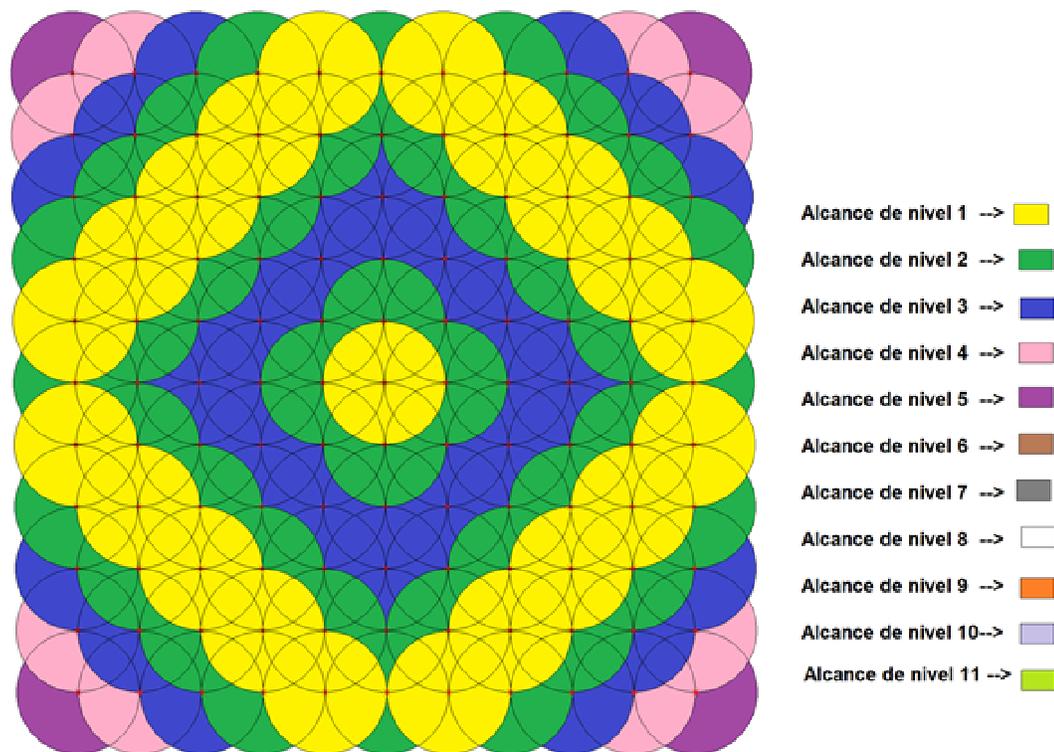


Figura 4.8: Ilustración de distribución de niveles del algoritmo de agrupamiento y mezcla de datos para una red que usa $max_cluster_level = 3$

de selección de ruta de datos mezclados” múltiplo de $Max_cluster_level \times 2$ como cabezas de conglomerado.

En caso de que una cabeza de conglomerado sea capaz de escuchar a otra cabeza de conglomerado con mayor disponibilidad de carga que la propia y esta tenga una trama por transmitir de tamaño menor que “*limite_de_mezcla*” esta retransmitirá la trama hacia la cabeza de conglomerado con mayor disponibilidad de energía para que sea mezclada con la información por transmitir de esta trama y reducir aun más el tráfico en la red. Es importante destacar que para reducir la latencia de las tramas se establece que las tramas solamente pueden ser retransmitidas una vez. Dicho de otra manera: una vez que una cabeza de conglomerado recibe una trama retransmitida, esta solamente puede ser mezclada con los datos en la cola de dicho conglomerado y transmitida hacia la base usando el protocolo de selección de ruta de datos mezclados. No se puede retransmitir nuevamente.

En la figura 4.8 se muestra la manera en que se espera se distribuyan los niveles del algoritmo de agrupamiento y mezcla de datos. En caso de que un nodo tenga varios vecinos de nivel inferior en *AMD* usará para retransmitir al que tenga menor nivel en *ADM*.

Algoritmo de compresión de datos: Existen en la literatura diversas técnicas de compresión de datos que se pueden utilizar, y no es el objetivo de la presente investigación el evaluar el impacto de las diferentes opciones disponibles por lo que se decide simplemente

agrupar el tipo de alarma, seguido por el número de nodos que reportaron la alarma y los identificadores; por ejemplo se asume que los nodos solamente tienen ocho bits de ID y que existen cinco alarmas posibles a transmitir en la red. Esto quiere decir que el campo de tipo de alarma solamente requerirá tres bits, no obstante se utilizarán ocho para dar la opción de agregar hasta 256 diferentes tipos de alarmas el del número de nodos que detectaron la alarma tendrá ocho bits y el de cada uno de los identificadores de nodo tendrá ocho bits respectivamente. En caso de que una cabeza de conglomerado reciba el reporte de cuatro alarmas tipo uno de los nodos 01h, 03h, 05h y 07h, y además reciba dos alarmas de tipo tres de los nodos 01h y 09h, los datos de la trama tendrían el siguiente formato:

01h	04h	01h	03h	05h	07h	03h	02h	01h	09h
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

En el algoritmo propuesto cada nodo tiene la capacidad de comprimir los datos recibidos y no solamente las cabezas de aglomerado. No obstante, al existir un mecanismo mediante el cual los datos son todos enviados a un punto específico, esto da más oportunidad para que los datos sean comprimidos-mezclados y se reducirá la cantidad de tramas transitando la red, reduciendo de esta forma la probabilidad de colisión.

4.3 Algoritmo multicapa propuesto

Sumando lo expuesto en las secciones 4.1 y 4.2; el algoritmo es descrito en la figura 4.5 y a continuación se detallan los estados que tienen diferencias con respecto de lo expuesto en la sección donde se explican los estados del *ADM*:

Descubrimiento de nivel de pertenencia: En esta etapa se hará lo mismo que se explicó anteriormente solamente que el nodo descubre su nivel de pertenencia en la red tanto para *ADM* como para *AMD*; para encontrar su nivel de *AMD* simplemente se observa el nivel de *ADM* del nodo una vez que se obtiene, si el nivel es múltiplo de $max_cluster_level \times 2$ este se ubicará como cabeza de grupo, o sea en el nivel uno de *AMD*, en caso contrario el nodo calculará la diferencia entre nivel de *ADM* del nodo y el múltiplo de $max_cluster_level \times 2$ más cercano y se ubicará en el nivel $diferencia + 1$. Por ejemplo: si $max_cluster_level$ es igual a tres, y el nivel de *ADM* del nodo es igual a siete, los múltiplos de $max_cluster_level \times 2$ más cercanos son seis y doce, con una diferencia de uno y cinco, por tanto el nodo tomará un nivel de *AMD* igual a dos.

Primera fase de petición de trama: En este estado el nodo envía una trama preguntado a los nodos de nivel superior si tienen una trama para retransmitir (de aquí en adelante se le llamará a esta trama *PT* “petición de transmisión”). En caso de no recibir respuesta, el nodo verifica si hay alarmas en la pila de alarmas y/o tramas en la pila de tramas (la diferencia entre una trama y una alarma es que las tramas se re-transmiten usando *ADM* mientras que las alarmas se retransmiten usando *AMD*). En caso de existir tramas y alarmas, se procede a comprimir las alarmas y las tramas en un solo paquete hasta alcanzar el tamaño máximo permitido por el protocolo. En caso de que la trama

resultante exceda dicho tamaño se crearán dos o más tramas para acomodar todos los datos en una cantidad mínima de tramas. Una vez que los paquetes resultantes están listos, se guardan en la pila de tramas y se pasa al primer estado de espera. En caso de que solo existan alarmas, estas se comprimirán y se enviarán usando *AMD* en el primer estado de espera. En caso de que no hayan ni tramas ni alarmas en la pila se retornará al estado de hibernación.

4.4 Diseño de las tramas usadas en el protocolo

Para el diseño de las tramas se procura que las mismas tengan una cantidad mínima de bits, esto con el propósito de minimizar el tiempo que duran las transmisiones y reducir la probabilidad de colisión así como el consumo de energía en el proceso de envío.

Cada uno de los espacios de información dentro de las tramas tendrá una longitud predefinida, con excepción de las tramas de alarma, las cuales como se mencionó en la sección 4.2.2, tendrán una longitud variable.

Para efectos del presente ejemplo se asume un byte para representar la mayoría de los campos en las tramas por simplicidad, no obstante estos tamaños pueden ser modificados para mejorar las características de la red en una aplicación específica, por ejemplo si se sabe que solamente se tendrán 8 tipos de tramas en la red entonces no necesario utilizar 8 bits para la cabecera de las tramas sino solamente 3 bits.

Las tramas requeridas para la implementación del protocolo anteriormente descrito son las siguientes:

Trama de *PT*: Esta trama contiene el nivel del nodo emisor con el objetivo de que los nodos que la escuchen puedan verificar que la misma corresponde a un nodo más cercano a la base que ellos mismos. Además la trama debe contener la dirección *MAC* del emisor para que el receptor sepa a quién debe hacer el *RTS*.

F1h Nivel_delEmisor_AMD Nivel_del_emisor_ADM Dirección_MAC_delEmisor

Trama de *RTS*: La trama de *RTS* debe contener la dirección *MAC* del emisor para que el nodo que la escuche la incluya en la trama del *CTS* y este sepa que su solicitud fue aceptada. Además, debe contener el nivel del nodo emisor para facilitar el proceso de búsqueda de identidad de los nodos vecinos y el largo de la trama que se desea transmitir para que en caso de ser escuchado por algún nodo vecino, este sepa el periodo por el cual debe esperar antes de intentar enviar su trama.

F2h Nivel_del_emisor_AMD Nivel_del_emisor_ADM Tamaño_trama_por_enviar
Dirección_MAC_delEmisor

Trama de *CTS* : La trama *CTS* es una trama de difusión y por esto debe contener la

dirección *MAC* del nodo a la que va dirigida para que los demás nodos identifiquen a quien se dirige y guarden silencio en caso de que no sea para ellos

F3h Nivel_emisor_AMD Nivel_emisor_ADM Tamaño_trama_por_enviar
Dirección_MAC_destinatario

Trama de alarma: Esta trama debe dejar claro el lugar de procedencia y su tipo. Es por esto que debe contener el código de la alarma así como la dirección *MAC* del nodo en el que se produjo. Además debe estar clasificada si debe ser enviada usando *ADM* o *AMD*. En caso de que el nodo tenga el reporte de cuatro alarmas tipo uno de los nodos del 01h, 03h, 05h y 07h, y además tenga en cola dos alarmas de tipo tres de los nodos 01h y 09h, los datos de la trama tendrían el siguiente formato:

F4h	01h	04h	01h	03h	05h	07h	03h	02h	01h	09h	F4h	CHECK_SUM
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----------

Código del tipo de alarma	Alarma asignada
0x04	Salida de operaciones del nodo
0x03	Fuego
0x02	Motosierras
0x01	Disparos
0x00	Inicio de operaciones del nodo

Tabla 4.1: Descripción de los tipos de alarmas asignadas hasta el momento

El campo que se asigna al código de la alarma es de un byte dando de esta forma la capacidad de comunicar 256 diferentes tipos de información, esto simplemente para dar espacio para la adición de alarmas nuevas en caso de ser necesario. Se definen cinco tipos de alarmas por el momento, las cuales se muestran en la tabla 4.1.

La salida de operación del nodo se define como una alarma generada por el módulo de alimentación e indica que el nodo se encuentra en un límite de carga que hace necesario apagar el nodo para que se cargue hasta un nivel que permita su operación normal o si no es posible recargar el nodo entonces para informar a la base que el nodo saldrá de operación debido a carga insuficiente. Para indicar si la alarma debe transmitirse usando *ADM* o *AMD* se utilizarán 2 cabeceras diferentes: F4 para *ADM* y F6 para *AMD*.

Trama de ACK: Esta trama debe dar la capacidad al nodo de saber si el dato fue recibido correctamente.

0xF5 CHECK_SUM Dirección_MAC_de_la_procedencia

Se añade un campo para un *CHECK_SUM* que para la implementación propuesta tendrá 8 bits y será simplemente los últimos 8 bits de la suma de los bits individuales del campo

de la información de la alarma, o sea, desde el byte de cabecera hasta antes del byte de *CHECK_SUM*. Se selecciona el anterior código de *CHECK_SUM* por simplicidad, no obstante queda al criterio del implementador del mismo el uso de algún otro método para validar que el dato se recibió correctamente.

4.5 Diseño de los tiempos del protocolo

A continuación se muestra un ejemplo de cómo definir los tiempos de espera e hibernación asociados protocolo propuesto, para una capa física en específico.

Para definir los tiempos del protocolo es necesario tener alguna referencia en cuanto al consumo energético que un nodo de comunicación tendrá en escucha pasiva, transmisión y en estado de hibernación, además de otros parámetros básicos que serán dependientes de la capa física sobre la cual se implemente la red. Dado que no se han definido los nodos físicos que se utilizarán para la implementación del presente proyecto en la red de sensores de monitorización del bosque, y que para efectos del diseño del protocolo se pretende obtener resultados independientes de la capa física, se tomará como referencia el consumo energético de los módulos TR1000 usados también como referencia en [36]. Los datos de consumo de potencia obtenidos de estos nodos se utilizarán solamente para obtener datos de referencia que posteriormente deberán ser repetidos usando datos que correspondan con la capa física que se quiera usar para la implementación de la red.

La base de tiempo B: Este es el tiempo que se utilizará como base para el cálculo del resto de los tiempos y es igual a la mitad del tiempo necesario para que la trama más grande del protocolo sea transmitida a la máxima distancia de alcance de la red, que sea escuchada y retransmitida hacia la primera estación nuevamente. Es importante mencionar que este tiempo es directamente dependiente de la capa física que se utilice en la implementación del protocolo, por lo que será necesario medirlo cada vez que se cambie la tecnología utilizada en esta capa. Para el caso de los nodos de referencia descritos anteriormente este tiempo es de aproximadamente 58ms; esto sabiendo que la trama potencialmente más larga del protocolo es la trama de alarma. Asumiendo arbitrariamente que el parámetro de red "*Limite_demezcla*" es igual a 10, o sea que la cantidad máxima de alarmas que se pueden mezclar en una trama son 10, la trama más larga posible en el protocolo será de 19 bytes, o sea, 4 bytes del tipo de alarma, asumiendo que se presenten las 4 alarmas posibles, 4 bytes para el número de alarmas de cada tipo que hay representadas en la trama, 1 byte para el encabezado y 10 bytes para los identificadores de los nodos que emitieron cada una de las alarmas.

Tiempo de hibernación: Este tiempo influye de manera directa en la cantidad de energía consumida en la etapa de comunicación de los nodos. Entre más grande sea el tiempo destinado en este estado, mayor será el ahorro en la energía del protocolo, no obstante el atraso en el proceso de envío de las tramas aumentará.

El consumo promedio de potencia de los módulos de comunicación se describe en la tabla

Estado	Potencia promedio
Escucha	12,5 mW
Transmisión	14,8 mW
Sueño profundo	0,016 mW

Tabla 4.2: Potencia promedio del TR1000 en los diferentes estados de la etapa de comunicación. (Tomada de [36])

4.2. Tomando en cuenta que en un protocolo en el que mientras la red se encuentra en reposo el módulo de comunicación se encuentra en estado de escucha se tiene que el consumo de potencia en reposo es constante, y la potencia promedio será de 12,5 mW.

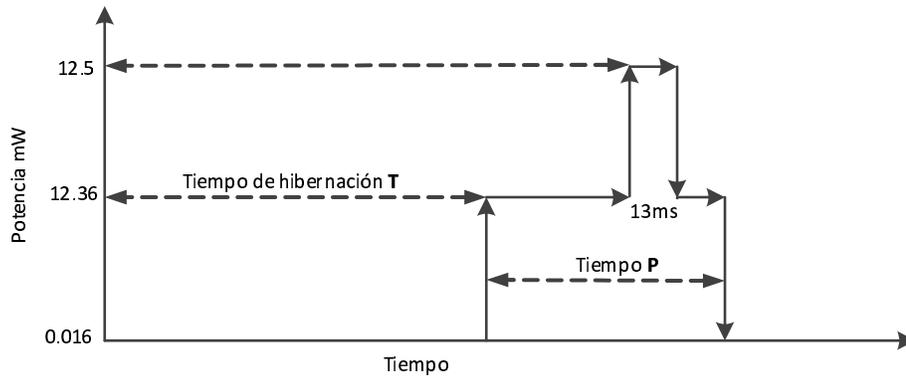


Figura 4.9: Gráfico aproximado del consumo de potencia esperado de los nodos con la red en reposo utilizando el protocolo propuesto

En el caso de la red en reposo con el protocolo implementado el consumo de potencia no será constante debido a que el módulo de comunicación pasará por diferentes estados a lo largo de cada uno de los ciclos de petición de trama e hibernación. De esta manera el consumo de energía en cada uno de los ciclos puede describirse aproximadamente como se muestra en la figura 4.9 donde la potencia promedio es:

$$P_{promedio} (mW) = \frac{0,016 \times T + 12,5 \times (P - 0,013) + 14,88 \times 0,013}{T + P} \quad (4.1)$$

De lo anterior se puede ver que la potencia consumida será dependiente de la proporción entre los tiempos P y T , si la red se encuentra en reposo (sin recibir señales de alarma) y se gasta un tiempo máximo aproximado de P ($2B$) en la primera fase de petición de trama, el porcentaje de ahorro de energía en el nodo con respecto de un protocolo en el que la etapa de radio (transmisión, recepción) debe mantenerse permanentemente encendida, se comportará aproximadamente como se muestra en la Figura 4.10 conforme aumenta el tiempo de hibernación

Se puede notar que el porcentaje de ahorro en la energía no cambia de manera drástica después de que el tiempo de hibernación alcanza un valor de $10P$, donde se alcanza un ahorro energético de aproximadamente 90%.

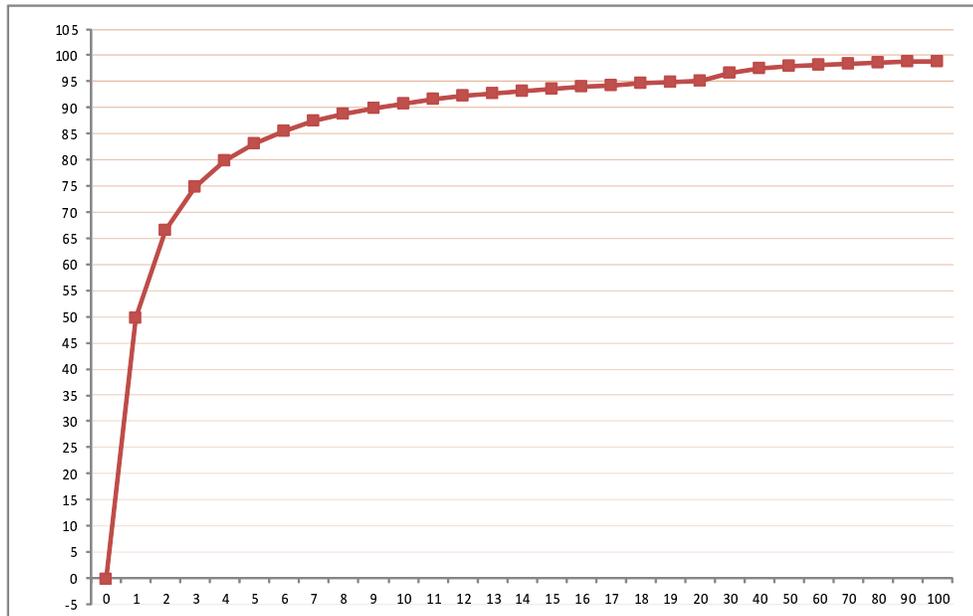


Figura 4.10: En el eje vertical se ve el porcentaje de ahorro en la energía para la red en reposo en la etapa de comunicación con respecto de un protocolo en el que esta etapa se mantiene constantemente encendida, mientras que en el eje horizontal se ve que tan grande es el tiempo T con respecto a P

El hecho de que el ahorro energético se ve acotado en aproximadamente 99,87% se debe a que para este caso la diferencia entre la potencia consumida en la etapa de sueño profundo y la potencia consumida en escucha pasiva es de 12,48mW lo cual es 99,87% de la potencia consumida en estado de escucha, de forma que conforme aumente el tiempo de hibernación la potencia consumida en el nodo tenderá a la potencia consumida con la etapa de transmisión apagada por lo que el ahorro máximo del protocolo con respecto de un protocolo en estado que no apaga su etapa de radio, será menor que de 99,87% para el presente ejemplo.

Para una red con N niveles en los que se programa un tiempo de hibernación T y la suma de todos los demás tiempos de espera es K , sin retransmisiones se tendrá que pasar como máximo por un proceso de agrupamiento de datos hacia una cabeza de conglomerado. En el peor de los casos se debe sumar al número de saltos $Max_cluster_level - 1$, tomando en cuenta que cada uno de los nodos opera sin sincronización. El tiempo máximo (en el peor de los casos) entre la detección de una alarma y su recepción en la estación base será:

$$T_{max} = ((N - 2) + (Max_Cluster_Level - 1)) \times (T + K) + T + T_{base} \quad (4.2)$$

Donde T_{max} es el tiempo de latencia máximo y T_{base} es el tiempo de ciclo base. Dado que el tiempo más grande contemplado en la red es el de hibernación, se tiene que el tiempo entre la emisión de una alarma y su recepción en la estación base será claramente afectado por el mismo.

Descubrimiento de nivel de pertenencia: El tiempo que se escucha en este estado debe ser lo suficientemente largo como para que el nodo escuche las tramas PT de todos sus vecinos y al mismo tiempo no debe ser más largo de lo necesario para que no se desperdicie energía.

Si se supone que la red se encuentra en reposo y ya ha sido configurada, entonces cada uno de los nodos transmitirá una trama PT aproximadamente cada vez que se termina el tiempo de hibernación; para asegurar que los nodos tengan el tiempo suficiente para escuchar las tramas PT de todos sus vecinos se asigna en este estado un tiempo de espera máximo del doble del tiempo de hibernación, o sea $2T$, en caso que después de $2T$ el nodo no escuche ninguna trama de PT entonces el nodo pasará al estado de hibernación y volverá al estado descubrimiento de nivel de pertenencia al terminar la hibernación.

Primer estado de espera: En este estado, al igual que en el estado de descubrimiento de nivel es necesario esperar tiempo máximo suficiente para escuchar las PT de sus vecinos o sea $2T$ si después de este tiempo no se escucha ninguna trama de PT válida, entonces se pasará al estado de descubrimiento del nivel de pertenencia.

Segundo estado de espera: En este estado se espera un tiempo aleatorio con el propósito de evitar colisiones después de haber escuchado una trama de PT de un nodo de nivel inferior. El tiempo aleatorio que se espera antes de enviar la trama debe ser lo más corto posible para que no se gaste mucha energía, no obstante entre más sean los posibles tiempos de transmisión en este estado, menor será la probabilidad de colisión .

Este estado influye directamente en el tiempo que debe ser asignado en la espera entre cada una de las tramas de PT de la primera fase de petición de trama y con esto influye directamente en el tiempo que debe ser asignado al estado de hibernación y al tiempo de retraso total de las tramas transmitidas.

Tomando en cuenta todo lo anterior este tiempo se define entre cero y $8B$, de esta forma el nodo tiene cinco tiempos posibles en los cuales puede intentar transmitir su trama de RTS ; en $t = 0$, $t = 2B$, $t = 4B$, $t = 6B$ y $t = 8B$ esto con el objetivo de que si otro nodo transmite en un tiempo menor el nodo que se encuentra esperando tenga tiempo suficiente para escuchar la trama de CTS y evite la colisión.

Primera fase de petición de trama: En este estado es necesario implementar un tiempo de espera antes de transmitir la trama de PT para evitar que la trama corrompa alguna transmisión en proceso por no haber escuchado otras tramas de PT , CTS o RTS . Este tiempo debe ser aproximadamente $2B$. Una vez que se transmite la trama de PT la estación debe esperar por $9B$ en el peor de los casos para dar tiempo a que la estación que desea reenviar su alarma transmita su RTS . Se puede notar que el tiempo P anteriormente usado para modelar el ahorro energético utilizando estados de hibernación será de aproximadamente $11B$ más el tiempo requerido para transmitir la trama de PT . En el caso de los nodos de referencia, el tiempo requerido para la transmisión de la trama será de aproximadamente 4,5ms, lo cual es despreciable si se compara los 58ms de la base de tiempo B por lo que de aquí en adelante se dirá que el tiempo P es de $11B$.

Estado de escucha: En este estado se espera un tiempo máximo de $2B$ después de haber enviado la trama de CTS para recibir el mensaje del nodo al que se le notificó, o para volver al estado de hibernación.

Verificación: En este estado se espera un tiempo máximo de $2B$ antes de dar por aceptado como correcto el mensaje recibido después de enviar la trama de ACK

Tercer estado de espera: Al igual que en el caso anterior el nodo esperará $2B$ para recibir un CTS .

Cuarto estado de espera: En este estado el nodo espera un tiempo $2B$ para recibir la trama de ACK .

4.6 Análisis teórico del protocolo propuesto

4.6.1 Retraso de las tramas suponiendo que no hay colisión

Anteriormente se dijo que el retraso máximo que sufriría una trama estaba dado por (4.2). Tomando en cuenta la información teórica suministrada en el apartado de diseño de los tiempos de la red, el tiempo de retraso para que una trama sea transmitida de un nivel al siguiente estará dado por:

- El tiempo transcurrido desde que se da la alarma y hasta que el nodo se despierte para atenderla. Este tiempo puede ser máximo de T y mínimo de 0.
- El tiempo transcurrido entre que el nodo entra al primer estado de espera y que escuche una trama de PT , este tiempo puede ser máximo de $T + 3B$ y mínimo de 0. Esto porque en el peor de los casos tiene que esperar a que el nodo siguiente salga del estado de hibernación un tiempo T , luego el nodo va a monitorizar el ambiente por un tiempo de $2B$ para ver que no hayan tramas siendo transmitidas y luego la trama puede tardar B para llegar al nodo que espera en el primer estado de espera
- El tiempo aleatorio seleccionado en el segundo estado de espera para enviar la trama de RTS puede ser entre $0B$ y $8B$.
- El tiempo necesario para transmitir la trama de RTS (aproximadamente igual a B).
- El tiempo que tarda en escuchar la trama CTS (aproximadamente igual a B).
- El tiempo que tarda en transmitir la trama de información (aproximadamente igual a B).
- El tiempo que tarde en escuchar la trama de ACK (aproximadamente igual a B).
- El tiempo que tarda el nodo que recibió la alarma en enviar otra trama de PT para verificar si hay más tramas de nodos de nivel superior que deseen enviar tramas a la base (aproximadamente igual a P que es igual a $11B$ asumiendo que no hay respuesta de otros nodos).

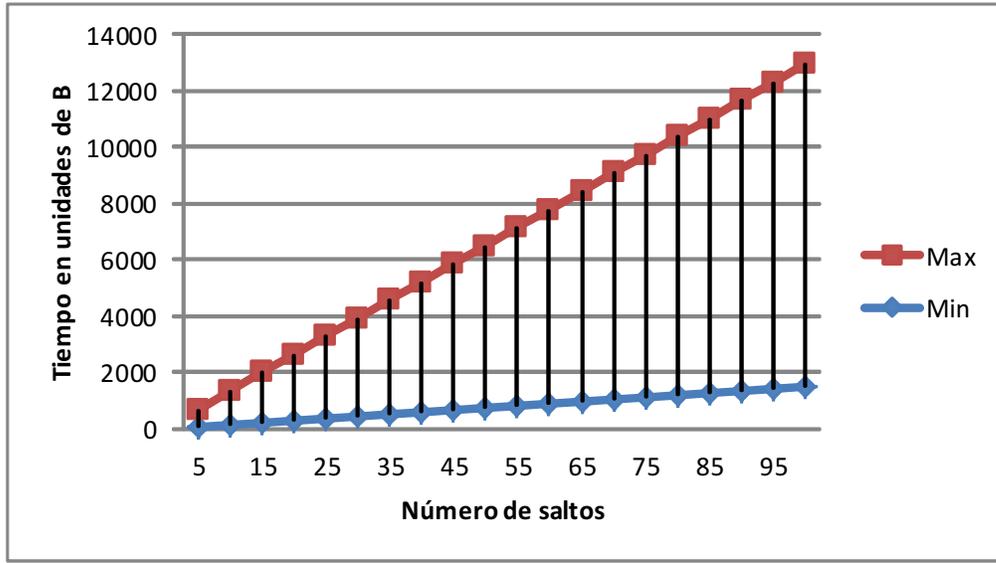


Figura 4.11: Ilustración del tiempo de retardo mínimo y máximo en unidades de B que tarda una alarma que se da en un nivel dado en llegar a la base asumiendo que no hay contención y que $P = 11B, T = 4P$ y $Max_Cluster_Level = 3$

$$T_{max} \approx 2T + 15B + P \quad (4.3)$$

$$T_{min} \approx 4B + P \quad (4.4)$$

Donde T_{max} y T_{min} son los tiempo máximos y mínimos de latencia. Esto excepto en la transición entre el nivel dos y el nivel 1 (o sea la base) en la que el retraso máximo será de $11B$ y el mínimo de $0B$. Esto quiere decir que el tiempo de espera en caso de no haber colisiones para un mensaje que viene de un ndo con nivel de ADM de N varía entre un rango máximo y mínimo como el que se muestra a continuación:

$$T_{max} \approx ((N - 2) + (Max_Cluster_Level - 1))(2T + 15B + P) + T + 11B \quad (4.5)$$

$$T_{min} \approx ((N - 2) + (Max_Cluster_Level - 1))(4B + P) \quad (4.6)$$

Esto quiere decir que para una red en la que se detecta una alarma en el nivel 50, con un tiempo B de 0,058s un $Max_cluster_Level$ de tres y asumiendo un tiempo $T = 4P = 44B$ el retraso máximo que sufrirá la trama en llegar a la base será de aproximadamente 14 minutos, y el mínimo de aproximadamente 9,6 minutos.

Este retraso es aceptable tomado en cuenta que la atención de la alarma después de recibida podría tomar horas. De las formulas anteriores se puede ver que este tiempo es proporcional a la base de tiempo B que dependerá de la capa física de la red.

En la figura 4.11 se muestran los valores de tiempo en unidades de B , de retraso máximo y mínimo para alarmas que ocurren en diferentes niveles basados en 4.5 y 4.6.

De este gráfico se puede notar que la brecha entre el valor máximo y el mínimo de retraso aumenta de forma lineal. Esta característica hace que entre más alto sea el nivel de la red en el que se produce un evento sea más difícil estimar el tiempo exacto en el que se produjo el mismo (asumiendo que el paquete de alarma no contiene hora del evento) además para una trama que se produzca en el nivel 50 se tendrá un retraso Máximo de $5755B$ y uno mínimo de $750B$ si $B = 0,058s$ y asumiendo como se hizo en el gráfico que $P = 11B, T = 4P$ y $Max_Cluster_Level = 3$ esto significaría un retardo máximo de 5,56 minutos y mínimo de 0,725 minutos.

En el siguiente apartado se busca proveer un modelo probabilístico para expresar este retardo.

4.6.2 Análisis probabilístico del retraso de las tramas con la red en reposo

Como se mencionó anteriormente, el tiempo de retraso para que una trama sea transmitida de un nivel al siguiente estará dado por:

$$T = A + B + C + D + E + F + G + H$$

En donde:

- A es el tiempo transcurrido desde que se da una alarma hasta el nodo se despierta para atenderla puede ser entre cero y T .
- B es el tiempo transcurrido entre que el nodo entra al primer estado de espera y que escuche una trama de PT . Este tiempo puede ser máximo de $T + 3B$ y mínimo de cero.
- C es el tiempo aleatorio seleccionado en el segundo estado de espera para enviar la trama de RTS . Puede ser entre $0B_{mín}$ y $8B_{máx}$.
- D es el tiempo necesario para transmitir la trama de RTS (aproximadamente igual a B).
- E es el tiempo que tarda en escuchar la trama CTS (aproximadamente igual a B).
- F es el tiempo que tarda en transmitir la trama de información (aproximadamente igual a B).
- G es el tiempo que tarda en escuchar la trama de ACK (aproximadamente igual a B).

- H es el tiempo que tarda en verificar que no hay más tramas por transmitir de los niveles superiores (aproximadamente igual a $P = 11B$)

Análisis de los tiempos A y B : Se llamará T al tiempo asignado al periodo de hibernación, y se llamará P al tiempo que se gasta en el proceso de petición de trama. Cuando la red se encuentra en reposo cada nodo envía una trama PT periódicamente cada $T + P$ segundos (exceptuando la base que lo hace cada P). De esta manera se tiene que para una red en reposo se puede decir que la probabilidad de que un nodo salga del estado de hibernación y envíe una trama PT en un periodo de tiempo de $T + P$ segundos es igual a 1. De la afirmación anterior y asumiendo una densidad de probabilidad uniforme, se deduce que la función de probabilidad de que un nodo salga de hibernación y envíe una trama de PT en un intervalo de tiempo dado entre a y b será:

$$\Psi = \int_a^b \frac{1}{T + P} dt \quad (4.7)$$

$$\Psi = \frac{1}{T + P} \times (b - a) \quad (4.8)$$

En donde la probabilidad de que el nodo salga del estado de hibernación en un periodo de tiempo dado entre a y b es igual a Ψ .

Análisis del tiempo C : A diferencia de los tiempos A y B , el tiempo C toma valores discretos y está diseñado de esta forma para evitar colisiones entre las tramas.

Si se supone que el generador de números aleatorios que se utiliza en la implementación del segundo estado de espera tiene una distribución constante, entonces la probabilidad de que el nodo seleccione cualquiera de las opciones de espera ($0B$, $2B$, $4B$, $6B$ y $8B$) es la misma e igual a 20%

Análisis de los tiempos $D-H$: Dado que estos tiempos son constantes, la consecuencia que tienen en la probabilidad del retraso en la transición de dos nodos de nivel superior a 1 es que establecen un tiempo mínimo de retraso de forma que la probabilidad de que el tiempo de llegada de la trama sea menos que $D + E + F + H = 0$

Probabilidad de retraso de las tramas en las transiciones de los nodos de nivel superior a 1 asumiendo que la red se encuentra en reposo: Tomando en cuenta que cada uno de los nodos trabaja de forma independiente, se pueden modelar los tiempos de espera de cada una de las transiciones como eventos independientes, de esta forma la probabilidad de retraso debida a los tiempos A y B se puede ver como una función de probabilidad uniforme continua como la que se muestra en la figura 4.12. Se parte de que los tiempos A y B son dos variables aleatorias independientes con funciones de densidad de probabilidad iguales a la que se muestra en la figura 4.12 y a las que llamaremos a y b , respectivamente. Entonces la densidad de probabilidad de la suma de los tiempos aleatorios $A + B$ vendrá dada por la convolución $a * b$.

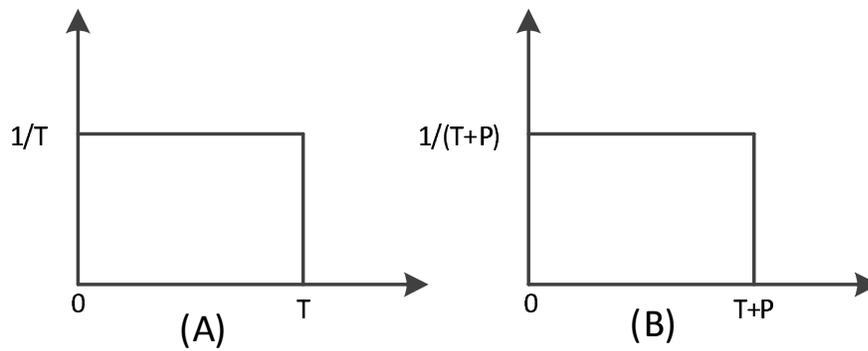


Figura 4.12: Función de probabilidad de los tiempos A y B

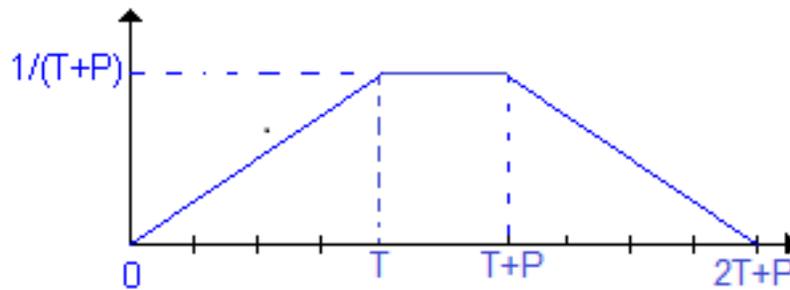


Figura 4.13: Función de probabilidad del retraso debido a $A+B$ sin tomar en cuenta el retardo constante debido a $D-H$

$$a * b = \int a(\tau) \times b(t - \tau) d\tau \tag{4.9}$$

Partiendo de lo anterior se sabe que la función de probabilidad que modela el retraso debido a los tiempos $A + B$ se puede modelar como se muestra en la figura 4.13.

Si se toma en cuenta el retraso aleatorio de C se tiene que la función de probabilidad será igual a la suma de las funciones de probabilidad para las diferentes selecciones posibles tomando en cuenta que cada una de las posibles opciones tiene 20% de probabilidad de ser seleccionada como se muestra en la figura 4.14.

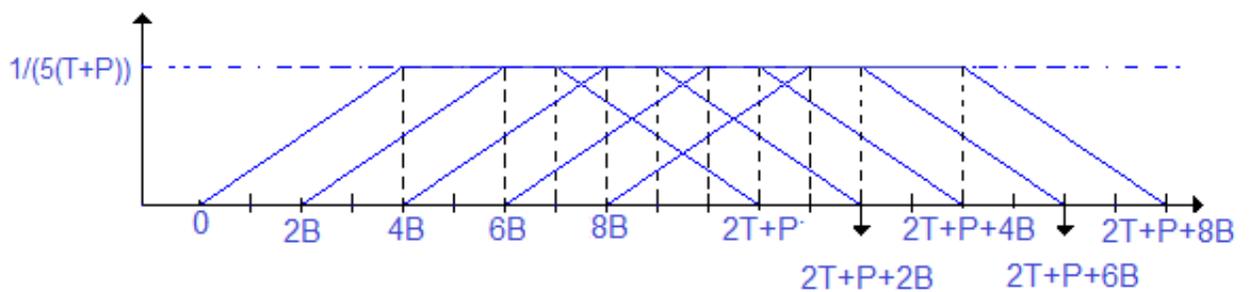


Figura 4.14: Funciones de probabilidad para el retraso para cada una de las posibilidades de selección de C

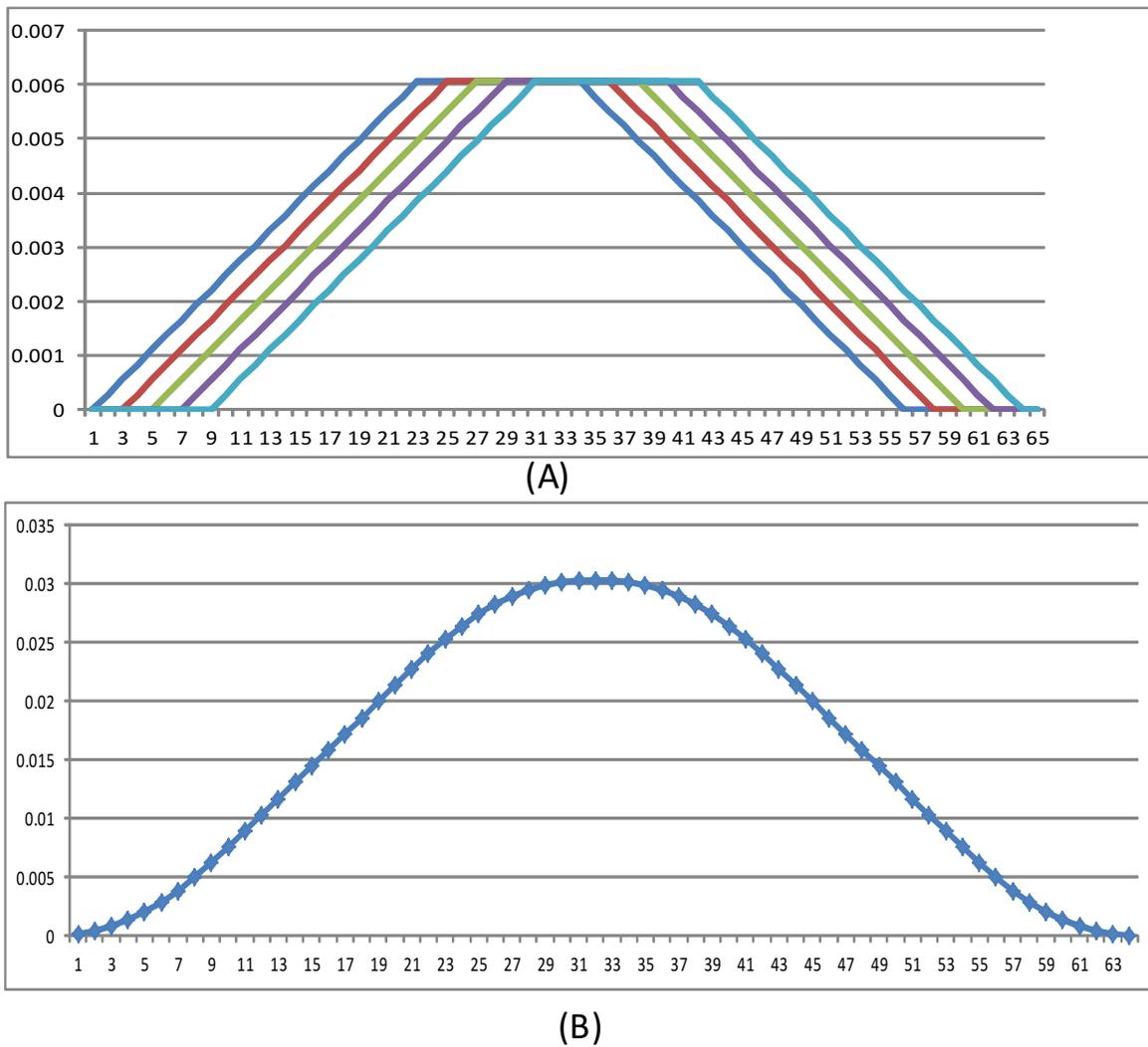


Figura 4.15: En (A) se muestra un ejemplo de las funciones de probabilidad independientes del tiempo C y en (B) se muestra la función de probabilidad que se obtiene al sumar el área debajo de las curvas

Para calcular la función de densidad de probabilidad para el tiempo de latencia de las tramas asumiendo que no hubo colisiones en la transmisión entre 2 nodos es necesario sumar las áreas bajo las curvas de las diferentes funciones de densidad de probabilidad como se hace en la figura 4.15 en donde se usa para el ejemplo $B=1$, $P = 11B$ y $T = 2P = 22B$

De la función anterior se puede calcular la probabilidad de que el retraso se encuentre en un intervalo dado entre a y b por medio de:

$$P_{ab} = \int_a^b f_x(X)dx \quad (4.10)$$

Intervalo	Probabilidad
$0B \leftrightarrow T$	0,2166
$T \leftrightarrow T+P+8B$	0,5668
$T+P+8B \leftrightarrow 2T+P+8B$	0,2166,

Tabla 4.3: Porcentaje de probabilidad de que el retardo de transmisión entre 2 nodos de nivel superior a 1 se encuentre en un intervalo dado

Se tiene que la probabilidad de que el retraso en la transmisión de las tramas se encuentre entre los diferentes intervalos posibles queda como se muestra en la tabla 4.3. Nótese que la tabla de probabilidades es independiente del tiempo B . Además se puede ver que la mayor probabilidad recae en el intervalo entre T y $T + P + 8B$ con un 56,7% de probabilidades, es importante además notar que en el análisis no se han tomado en cuenta los retardos debidos a los tiempos del D al H , dado que estos tiempos son constantes, simplemente se corre la curva obtenida de forma que no empiece en 0 sino en $15B$ que es el retardo mínimo correspondiente a estos tiempos. No obstante que las probabilidades en los diferentes intervalos sean independientes de la base de tiempo B conforme B disminuya el retraso en las tramas lo hará de una manera proporcional.

También es importante destacar que conforme el número de posibles valores del tiempo C aumente, el intervalo de valores posibles en el retraso aumentará hacia la derecha e igualmente lo hará la curva de probabilidades.

Si se quiere calcular la función de probabilidad de el retardo de las tramas para un nodo para más de un salto en la red es necesario hacer la convolución de la función anteriormente calculada para cada uno de los saltos.

Se puede notar en la figura 4.15(b) que la distribución de probabilidad resultante de la convolución de las distribuciones de probabilidad para las diferentes opciones de C resulta ser una función muy similar a una distribución normal con $\mu = (2T + P + 8B)/2 = T + P/2 + 4B$ este resultado es esperado debido al teorema del límite central, el cual establece que la suma de variables aleatorias independientes, idénticamente distribuidas y con varianza finita (como las mostradas en la figura 4.14) se distribuye aproximadamente como una normal[17].

Además se sabe que para una distribución normal X de media μ y varianza $\sigma^2 \geq 0$, es posible encontrar n variables aleatorias independientes X_1, \dots, X_n cada una con distribución normal de media μ/n y varianza σ^2/n dado que la suma $X_1 + \dots + X_n$ de estas n variables aleatorias $X_1 + X_2 + \dots + X_n$, tenga esta específica distribución normal. de lo anterior se puede concluir que la distribución de probabilidad para el retardo de una trama en uno de nivel n asumiendo que no hay colisiones sera una distribución de probabilidad normal con μ/n y varianza σ^2/n donde μ y σ van a ser la media y la varianza de la distribución de probabilidad de para la latencia de las tramas de un nivel al siguiente como se muestra en la figura 4.15(b).

4.7 Desarrollo del ambiente de simulación

4.7.1 Desarrollo de la Simulación del medio

Uno de los retos que es necesario resolver cuando se trabaja con un simulador basado en eventos como Omnet++ al simular redes inalámbricas de sensores, es el modelado del medio inalámbrico en sí mismo. En un medio inalámbrico se debe tomar en cuenta que la conexión entre los sensores va a depender de la potencia de transmisión del nodo que emite el mensaje, de la posición de los nodos, tanto los que escuchan como los que transmiten, y de la geografía en la que se encuentran ubicados, ya que es posible que existan obstáculos que no permitan la transmisión entre nodos colocados a una distancia pequeña. Además es necesario tomar en cuenta que es posible que exista colisión de tramas cuando alguno de los nodos que reciba una transmisión, se encuentre en el rango de escucha de otra transmisión en el mismo momento. El nodo emisor no necesariamente será capaz de reconocer que ocurrió una colisión.

Se propone para la simulación del medio de transmisión en el presente trabajo el uso de Mixim[33] Mixim es una biblioteca de modelado para Omnet++, creada para redes inalámbricas fijas y móviles (redes inalámbricas de sensores, redes vehiculares, etc). Ofrece modelos detallados de propagación de ondas, estimación de interferencias, consumo del tranceptor, y protocolos de acceso al medio. Una red MiXiM básica se compone de los siguientes elementos:

- Módulo de definición de la red
- Módulo de definición del nodo
- Módulo de interfaz de red, que define la capa de red del nodo
- La configuración de la capa física, modelo analógico (cálculo de atenuación) y *decider* (clasificación de ruido y cálculo de errores de bit).
- La configuración de la simulación

En la figura 4.16 se muestra un fragmento de código para la definición de un módulo de red en mixim. En la definición de red se incluyen parámetros de definición de las dimensiones del terreno, los submódulos de utilidad global y de administración de conexiones, y el tipo de nodo que compone la red. El administrador de conexiones verifica si dos nodos pueden oírse uno al otro y actualiza sus conexiones de acuerdo a eso. Si dos nodos están conectados significa que pueden recibir algo de cada uno, pero no significa que pueden entenderse. Si dos nodos no están conectados, están fuera del rango de alcance y no recibirán ninguna señal uno del otro. Se debe diferenciar el rango de alcance del modelo de comportamiento de la radio. El rango de alcance define la distancia a partir de la cual un nodo alejado deja de existir para el tranceptor. El modelo analógico y el *decider*, junto con los parámetros del módulo de interfaz de red, determinan la intensidad de señal combinada y si una trama es recibido o no.

El *decider* y los modelos analógicos se definen con un archivo xml que configura parámetros propios. En el módulo de interfaz se debe especificar qué archivo contiene esta configura-

```

package org.mixim.examples.baseNetwork;

import org.mixim.base.connectionManager.ConnectionManager;
import org.mixim.base.modules.BaseWorldUtility;

module BaseNetwork
{
    parameters :
        double playgroundSizeX @unit(m); // x size of the area the nodes are in (in meters)
        double playgroundSizeY @unit(m); // y size of the area the nodes are in (in meters)
        double playgroundSizeZ @unit(m); // z size of the area the nodes are in (in meters)
        double numNodes; // total number of hosts in the network

    submodules :
        connectionManager : ConnectionManager;
        world: BaseWorldUtility {
            parameters :
                playgroundSizeX = playgroundSizeX;
                playgroundSizeY = playgroundSizeY;
                playgroundSizeZ = playgroundSizeZ;
        }
        node[numNodes]: BaseNode;
}

```

Figura 4.16: Fragmento de código de un módulo de definición de red en Mixim (Tomado de [33]).

```

package org.mixim.examples.baseNetwork;

import org.mixim.base.connectionManager.ConnectionManager;
import org.mixim.base.modules.BaseWorldUtility;

module BaseNetwork
{
    parameters :
        double playgroundSizeX @unit(m); // x size of the area the nodes are in (in meters)
        double playgroundSizeY @unit(m); // y size of the area the nodes are in (in meters)
        double playgroundSizeZ @unit(m); // z size of the area the nodes are in (in meters)
        double numNodes; // total number of hosts in the network

    submodules :
        connectionManager : ConnectionManager;
        world: BaseWorldUtility {
            parameters :
                playgroundSizeX = playgroundSizeX;
                playgroundSizeY = playgroundSizeY;
                playgroundSizeZ = playgroundSizeZ;
        }
        node[numNodes]: BaseNode;
}

```

Figura 4.17: Fragmento de código del módulo *decider* en Mixim (Tomado de [33]).

ción.

En el fragmento de código en la figura 4.17 es un ejemplo del contenido del archivo xml. En este ejemplo, se declara un modelo analógico de tipo *Break – pointPathlossModel*, que representa el modelo de atenuación definido en el estándar IEEE 802.15.4. Como puede observarse, es posible declarar más de un modelo analógico. Luego se declara el tipo de *decider*, *Decider802154Narrow*, que clasifica la señal en bits o ruido, según el

modelo *BER* definido también en el estándar IEEE 802.15.4.

Módulo	Propósito
WSNRouting	Definición de red para simulación de redes inalámbricas de sensores
host802154_2400MHz	Definición de nodo que utiliza un tranceptor 802.15.4 a 2,4 GHz.
BatteryStats	Módulo de recolección de estadísticas sobre la batería.
SimpleBattery	Modelo simple de batería.
StationaryMobility	Administrador de información básica de movilidad y posición.
Nic802154_TI_CC2420	Modelo de interfaz de red Texas Instruments CC 2420 IEEE 892.15.4 CSMA.
CSMA802154	Enlace IEEE 802.15.4-2006 CSMA no ranurado
PhyLayerBattery	Capa física que consume batería

Tabla 4.4: Algunos de los módulos de Mixim usados en la simulación de la red

En la tabla 4.4 se enumeran algunos de los módulos de MiXiM utilizados para modelar la red de referencia.

Modelado de los Nodos

Los nodos de la red se modelan usando los siguientes submódulos:

battery: Permite definir una capacidad inicial de batería, y que la radio pueda consumir energía descontando de esa capacidad. **nic:** Define el comportamiento de la interfaz de red. Es a su vez un módulo compuesto por el módulo de capa de enlace y el módulo de capa física. El modelo de interfaz utilizado es IEEE 802.15.4-2006 y está implementado por el módulo ic802154 TI CC2420, descrito más adelante. **net:** Define el protocolo de red que utiliza el dispositivo. En este trabajo se desarrolló un nuevo módulo que incluye la capa de red, la capa de control de acceso al medio para el protocolo propuesto, mientras que para el nodo de referencia se utiliza el protocolo de red wiseroute. **app:** Define el comportamiento de la aplicación que utiliza los servicios de red. Para el caso de las pruebas realizadas en el presente proyecto solamente se usa esta capa para simular el tráfico que circula por la red.

Interfaz de red IEEE 802.15.4: El módulo Nic802154 TI CC2420 implementa una interfaz de red Texas Instruments CC 2420 802.15.4 usando el protocolo de enlace CSMA especificado en el estándar IEEE 802.15.4-2006. El modelo CSMA802154 fue validado independientemente en una red inalámbrica de sensores de prueba[34], aunque la validación

```
class PhyLayerBattery: public PhyLayer {  
  
    enum Activities {  
        SLEEP_ACCT=0,  
        RX_ACCT,  
        TX_ACCT,  
        SWITCHING_ACCT,  
        DECIDER_ACCT,  
    };  
  
    ...  
};
```

Figura 4.18: Fragmento de código del módulo de batería en Mixim (Tomado de [33]).

fue realizada con una cantidad de nodos muy pequeña.

Capa física: El módulo de capa física permite configurar la sensibilidad y potencia de transmisión. Para el tranceptor cc2024 la máxima potencia de transmisión es 1 mW [16] y esta es la que se utilizó en las simulaciones. No se ha podido hallar el alcance de la radio en la hoja de datos del tranceptor[16]. En la documentación del nodo MICAz de la compañía MEMSIC[21] el cual es basado en el mismo estándar y tiene características técnicas muy similares, se especifica que el alcance del tranceptor es de 75 a 100 m en exteriores, y de 20 a 30 m en interiores. Se configuró un valor de sensibilidad de -95 dBm [16]. El módulo de capa física utilizado calcula la atenuación utilizando el modelo establecido en la sección E.5.3 del estándar IEEE 802.15.4-2006[1]. Esta capa utiliza el módulo *decider* Decider802154Narrow para filtrar las señales recibidas según su intensidad, calcular los errores de bits y clasificar las señales en ruido o paquetes. Para las simulaciones realizadas se deshabilitó el ruido térmico.

Modelo de batería: La batería es un módulo independiente, y define cuál es la capacidad inicial y el voltaje. La capa física de la interfaz Nic802154 TI CC2420, PhyLayerBattery, descuenta energía de la batería según tipos de actividades definidas: dormir, recepción, transmisión, cambio de modo, y actividad del *decider*. El fragmento de código de la figura 4.18 muestra la definición de las posibles actividades en el módulo.

En general el modelo usado para las implementación de los nodos será como se muestra en la figura 4.19.

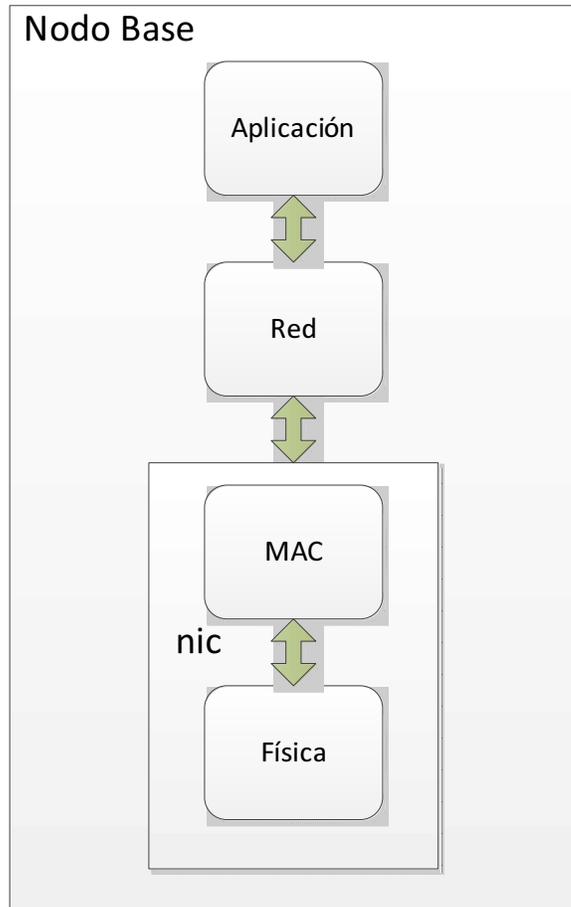


Figura 4.19: Módulos usados en la implementación de los nodos en Omnet++

Capítulo 5

Resultados y análisis

Para el diseño de las pruebas que se utilizarán en la caracterización del protocolo propuesto en el presente trabajo, se debe tomar en cuenta que los parámetros de latencia, consumo de potencia, tiempo de configuración y colisiones van a depender no solamente de los factores de configuración del protocolo sino también de factores externos al mismo como lo son: la topología física de la red, el tiempo de ocurrencia de las alarmas, los nodos que detecten las tramas en un área geográfica determinada, y los posibles fallos en los nodos.

Dado lo anterior, si se decide por ejemplo probar la latencia las tramas usando mil escenarios diferentes en los que se usan parámetros aleatorios de frecuencia de generación de las alarmas, topología, nodos de ocurrencia de las alarmas y cantidad de nodos en la red; se van a tener diferencias en los resultados de un escenario al siguiente, debido a que el cambio de alguno de los parámetros anteriormente descritos puede causar cambios drásticos en las condiciones de la red, por ejemplo crear cuellos de botella para el envío de mensajes por algún nodo en particular entre muchos otros casos posibles, y la cantidad de configuraciones posibles para probar es infinita. Es por esto que dependiendo del parámetro de la red que se desee caracterizar se van a seleccionar parámetros específicos que se consideren más relevantes para modificar en cada uno de los escenarios, y no todos al mismo tiempo, de manera que se pueda obtener información relevante para cada uno de los casos.

Los principales parámetros que se desean caracterizar basados en los objetivos específicos que se plantearon al inicio de la presente investigación son:

- Consumo energético de los nodos de la red con respecto de los nodo en una red en la que se implemente el estándar IEEE802.15.4.
- Tiempo de latencia de las tramas
- Comparar los resultados de la simulación del consumo energético de los nodos y de la latencia de las tramas con respecto del modelado teórico expuesto anteriormente para estos parámetros
- Capacidad de autoconfiguración de la red ante fallos en los nodos.

Los parámetros utilizados para cada una de las pruebas excepto en los casos en los que

Parámetro de Mixim	Descripción	Valor
sleepCurrent	Corriente promedio del nodo en estado de hibernación	0,021 uA
rxCurrent	Corriente promedio del nodo en estado de escucha	17,4 mA
txCurrent	Corriente promedio del nodo en estado de transmisión	18,8 mA
setupRxCurrent	Corriente promedio del nodo mientras se pasa al estado de escucha	0,6391mA
setupTxCurrent	Corriente promedio del nodo mientras se pasa al estado de	0,6845mA
phy.headerLength	Tamaño del encabezado de la capa física	48 bit
phy.timeSleepToRX	Tiempo necesario para pasar de hibernación a escucha	0,001792 s
phy.timeSleepToTX	Tiempo necesario para pasar de hibernación a transmisión	0,001792 s
phy.timeRXToTX	Tiempo necesario para pasar de escucha a transmisión	0,000192 s
phy.timeTXToRX	Tiempo necesario para pasar de transmisión a escucha	0,000192 s
phy.timeRXToSleep	Tiempo necesario para pasar de escucha a hibernación	0 s
phy.timeTXToSleep	Tiempo necesario para pasar de escucha a hibernación	0 s

Tabla 5.1: Resumen de los parámetros de la etapa de radiofrecuencia usados por defecto en las pruebas

Parámetro de la Red	valor usado
<i>limite_de_mezcla</i>	100 alarmas
<i>B</i>	58ms
<i>T</i>	$44B = 4P$
<i>X</i>	10
<i>Max_cluster_level</i>	3

Tabla 5.2: Resumen de las parámetros del protocolo usados para las pruebas

así se especifique se muestran en las tablas 5.1 y 5.2.

A continuación se describe el procedimiento utilizado para cada una de las pruebas realizadas, así como los resultados obtenidos.

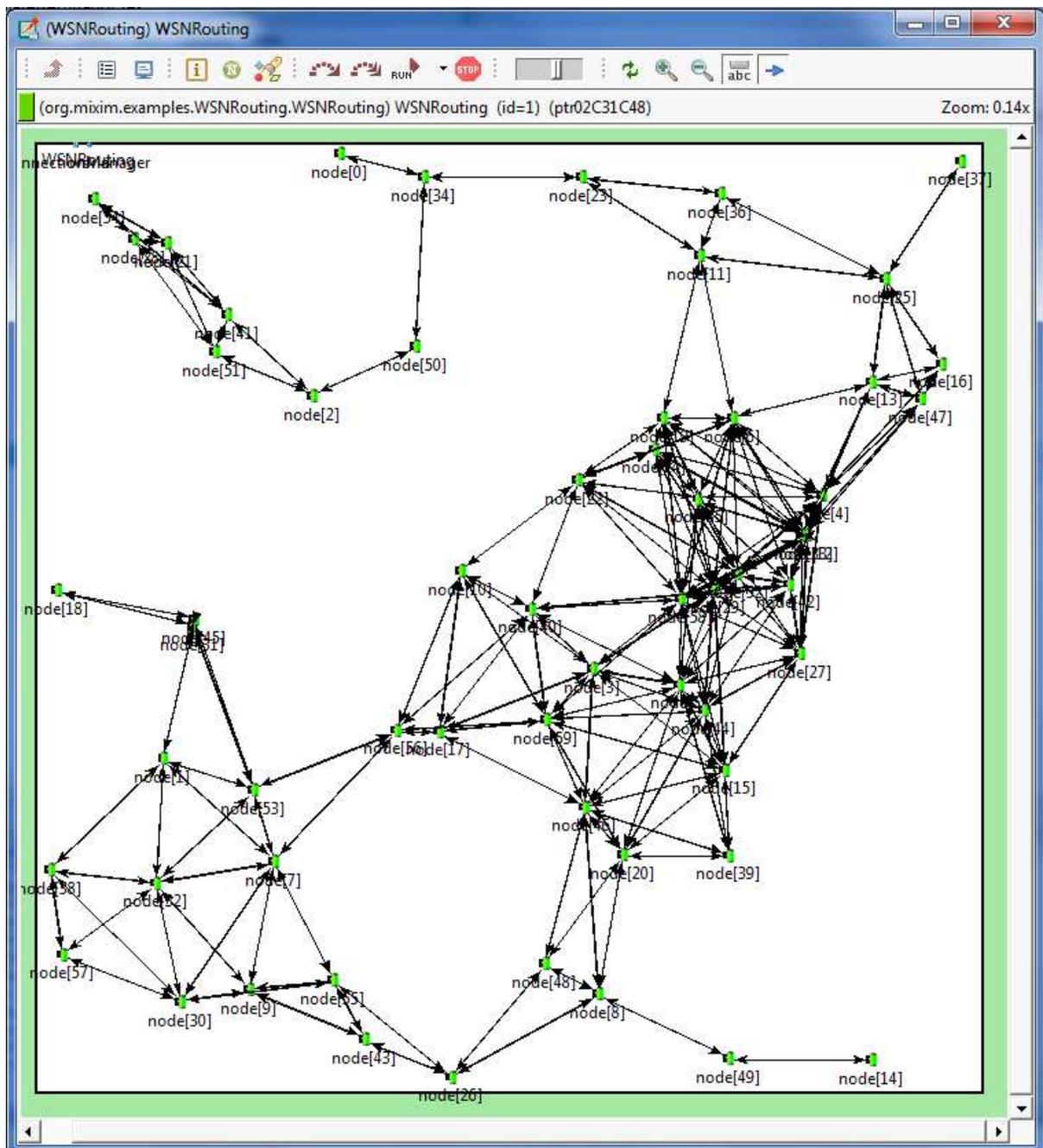


Figura 5.1: Configuración física de red para una de las pruebas de autoconfiguración realizadas

5.1 Capacidad de autoconfiguración de la red ante fallos en los nodos

En la primera prueba para comprobar la capacidad de autoconfiguración de la red, se crean cien escenarios en los que se ponen sesenta nodos de forma aleatoria en un área de simulación de $25000000m^2$ (5000m en X y 5000m en Y) como se muestra por ejemplo en la figura 5.1. Para todos los casos la base es el nodo[0].

Para la realización de esta prueba se configuró la red como se muestra en la tabla 5.2.

Tiempo (Minutos)	Evento
0	Inicio de la red
5	se apagan nodos 1 y 59
7	se apagan nodos 2 y 58.
9	se apagan nodos 3 y 57.
11	se apagan nodos 4 y 56.
13	se apagan nodos 5 y 55.
15	se apagan nodos 6 y 54.
17	se apagan nodos 7 y 53.
19	se apagan nodos 8 y 52.
21	se apagan nodos 9 y 51.
23	se apagan nodos 10 y 50.
25	se encienden todos los nodos de nuevo.
100	finaliza la prueba y se cuentan las alarmas que llegan al sumidero

Tabla 5.3: Tabla de resumen de la prueba de capacidad de autoconfiguración y efectividad en la entrega de tramas

En la prueba todos los nodos se encienden en tiempos aleatorios desde cero hasta dos minutos de la simulación, esto con el objetivo de aleatorizar el estado de los nodos de una forma similar a como sería en ambientes reales, y se empiezan a generar alarmas en todos los nodos cada minuto después de cinco minutos de encendida la red para dar tiempo a que los nodos lleguen a una configuración estable. En ese momento se empiezan a apagar las etapas de radio de algunos nodos como se muestra en la tabla 5.3; cuando se llega a veinticinco minutos de simulación se dejan de generar alarmas y se encienden todos los nodos de nuevo, la prueba finaliza a los cien minutos de simulación. Aún cuando los nodos se encuentran con la etapa de radio apagada, estos siguen generando y guardando alarmas, las cuales deberán ser enviadas cuando los nodos vuelvan a encender sus etapas de radio. Una vez que se termina la prueba, se cuentan las alarmas recibidas en el nodo base para corroborar que la red fue capaz de reconfigurarse para enviar las alarmas al sumidero.

En esta prueba específica se obtiene que llegaron al nodo base un total de mil docientas tramas de alarma de tipo nueve para cada uno de los escenarios, lo cual indica que los nodos fueron capaces de reconfigurarse para soportar los cambios de topología causados por los nodos que salieron de la red y después regresaron.

Además de los cien casos anteriormente mencionados se realizan cien pruebas más usando configuraciones como la que se muestra en la figura 5.1, en la que se repite la misma prueba anteriormente descrita, solo que esta vez no se enciende todos los nodos apagados en el minuto veinticinco sino que estos permanecen apagados y simplemente se cuentan

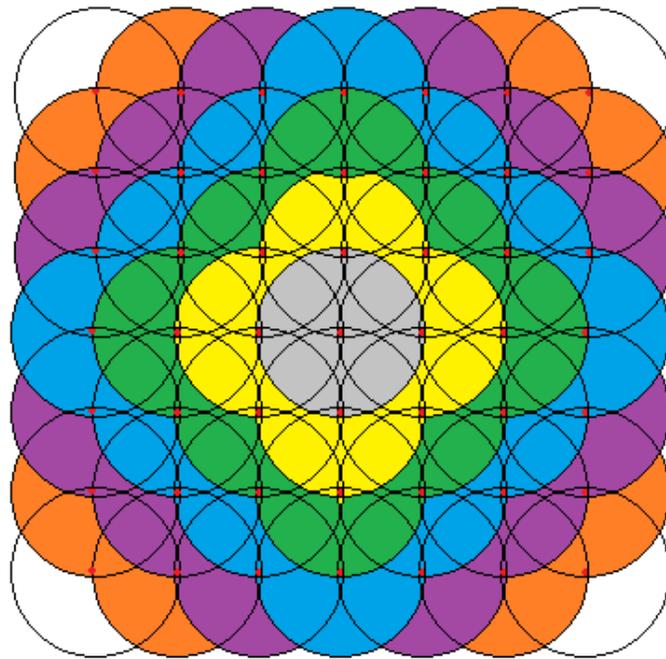


Figura 5.2: topología en cuadrícula para una red de sensores en la que los nodos se encuentran al límite del alcance de radio de sus vecinos

las alarmas que llegan a la base para verificar que la red fue capaz de reconfigurarse para tomar en cuenta los fallos en los nodos. La selección de estos nuevos cien casos se debió hacer muy cuidadosamente debido a que si se apaga un nodo que deje aislado un segmento de la red, dicho segmento no será capaz de enviar los datos a la base debido a que no existirán caminos posibles para hacerlo. Es por esta razón que se escogen topologías, en donde se sabe que los nodos que se apagan si bien es cierto requerirán de los nodos de la red cambien sus datos de nivel de ADM y AMD, se sabe que estos dejan la posibilidad de encontrar otro camino a la base, para estos casos se generan nodos en posiciones aleatorias pero luego se utiliza un programa para validar que las posiciones seleccionadas cumplen con la restricción anteriormente expuesta, si cumple se mantiene el escenario y si no se genera una nueva topología hasta encontrar los cien casos que funcionarán para la prueba.

En este caso se tiene que la cantidad de alarmas reportadas a la base fueron novecientas ochenta, lo cual era lo esperado para esta prueba, e indica que los nodos fueron capaces de reconfigurarse para soportar los cambios de topología causados por los fallos en los nodos.

Otro parámetro que se considera relevante tomar en cuenta con respecto a la capacidad de autoconfiguración de la red, es el tiempo que tardan los nodos de diferentes niveles en obtener su nivel de ADM. Para medir esto se utilizarán dos topologías base; en una se ubican los nodos en una cuadrícula como se muestra en la figura 5.2 solamente que en lugar de hacer la cuadrícula de 7X7 se hará de 51X51, la segunda topología será como la que se muestra en la figura 5.3 solo que en lugar de tener cinco niveles se tendrán cincuenta niveles. Para ambas topologías se harán dos experimentos los cuales se repetirán cien

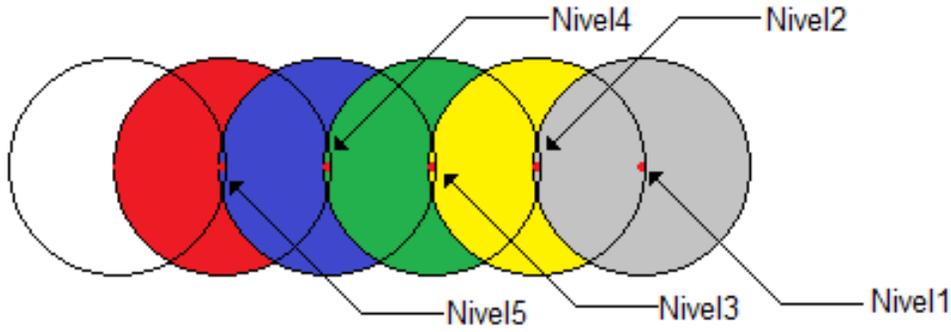


Figura 5.3: topología en línea para una red de sensores en la que los nodos se encuentran alineados al límite del alcance de radio de sus vecinos]

veces; en el primero se encienden todos los nodos en tiempos aleatorios del tiempo cero hasta dos minutos de la simulación, luego en el tercer minuto de la simulación se enciende la base y desde ahí se mide el tiempo que tarda cada uno de los nodos de la red en los diferentes niveles en obtener su nivel de *ADM* y *AMD*, en el segundo experimento una vez que todos los nodos tienen sus niveles configurados se cambia el nivel de la base para que en lugar de ser uno sea cinco y se mide el tiempo que tardan los nodos en reconocer el cambio con y sin tráfico en la red para diferentes valores del parámetro X , el cual se define como el número de veces que cada nodo pasa por el estado de hibernación antes de pasar al estado de descubrimiento de nivel para verificar su nivel. Este experimento se hace usando la topología lineal anteriormente descrita. Para la generación del tráfico se insertan alarmas en los nodos de nivel múltiplo de cinco cada minuto.

Nivel	Retardo promedio en topología lineal	σ^2 en topología lineal	Retardo promedio en topología malla	σ^2 en topología malla
5	272,68	681,41	254,30	398,07
10	599,56	1753,54	566,59	1133,88
15	935,32	2162,42	872,0947	1374,30
20	1268,60	3402,07	1181,67	1828,41
25	1602,87	3578,15	1490,07	2194,07
30	1935,67	4384,13	1798,99	2544,20
35	2269,48	5916,29	2105,93	3217,46
40	2604,41	6821,45	2419,39	3702,63
45	2931,20	7403,77	2726,38,	4009,99
50	3259,73	7925,22	3034,99	4691,99

Tabla 5.4: Datos de tiempo para la obtención de nivel de *ADM* y *AMD* en las topologías lineal y en malla para la red en reposo usando valores normalizados en términos de B

En la tabla 5.4 se muestran los resultados obtenidos para los tiempos promedio de adquisición del nivel de red normalizado en función de B cuando la red está en reposo. Se puede

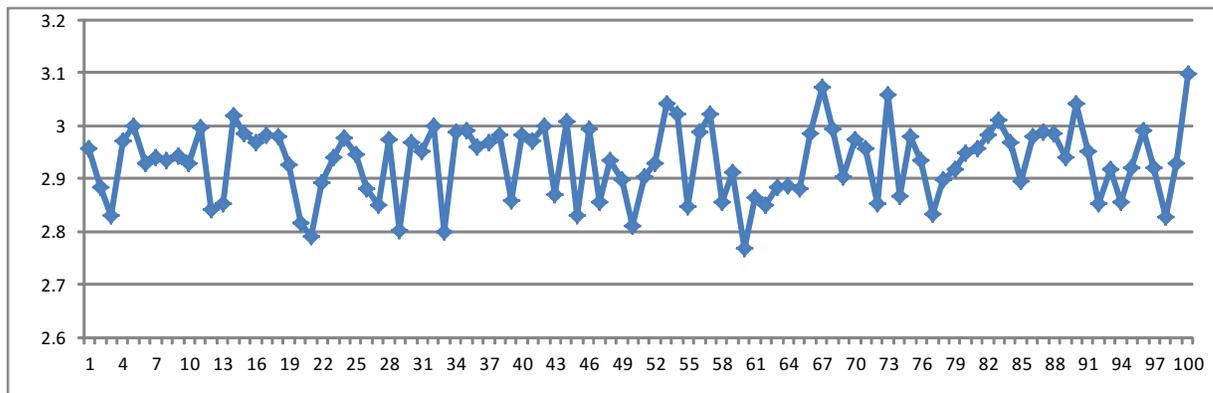


Figura 5.4: Retardo de la primera configuración de los nodos de nivel 50 para la topología de malla, en el eje Y se observa el retardo en minutos y en el eje X se observa el número de experimento

notar que los tiempos para la topología de malla son un poco menores a los tiempos para la topología lineal, este resultado puede deberse a que en la topología de malla los nodos pueden tener más de un vecino de nivel inferior, razón por la que hay mayor probabilidad de escuchar tramas de *PT* de algún camino que se configuró más rápido que otro, además se puede notar que la varianza de los datos aumenta con el nivel de los nodos de la red de manera a mayor nivel del nodo mayor es la variabilidad del tiempo necesario para que el nodo identifique su nivel de red. Para un nodo de nivel 50 el tiempo promedio que se necesitó para obtener su nivel de red en la topología de malla, tomando en cuenta un valor de $B = 0,058s$ fue de: $3035 * 0,058/60 = 2,93minutos$ en la topología de malla

En la figura 5.4 se observan los resultados de retardo para la obtención del nivel de red en los nodos de nivel 50 cuando la red se encuentra en reposo, en esta figura se pueden notar las diferencias de tiempo entre el retardo de adquisición de nivel entre diferentes experimentos.

En la tabla 5.5 se muestran los resultados obtenidos para los tiempos promedio de adquisición del nivel de red normalizado en función de B cuando se le añade tráfico a la red. Se puede notar que al igual que en el caso anterior el retardo para la configuración lineal es un poco mayor que para la de malla, dicho comportamiento se puede atribuir a que para transmitir las alarmas en la topología lineal todos los mensajes deben usar el mismo camino de forma que cada nodo tiene un solo vecino, no obstante en la topología de malla cada nodo tiene más de un vecino por lo que puede escuchar más ramas de *PT* por lo que es posible que el nodo se configure un poco más rápido.

En la tabla 5.6 se muestran los resultados de tiempo promedio requerido para detectar un cambio en la topología para nodos detectados a diferentes niveles de distancia del cambio con la red en reposo, en la tabla 5.7 se muestran los mismos resultados pero con tráfico en la red, se puede notar que cuando se usan los parámetros de la red como se muestra en la tabla 5.2, un nodo a 50 niveles de distancia de un cambio importante en su topología puede tardar en promedio 11 horas en darse cuenta, si la red está en reposo, mientras que

Nivel	Retardo promedio en topología lineal	σ^2 en topología lineal	Retardo promedio en topología malla	σ^2 en topología malla
5	289,38	1178,54	279,98	783,34
10	652,00	2124,38	617,15	1454,53
15	1006,13	4011,51	963,51	2488,13
20	1363,39	5405,66	1308,34	3170,50
25	1724,28	5633,33	1642,56	4438,31
30	2085,34	7049,61	1984,34	5405,02
35	2438,74	7925,81	2325,75	6272,56
40	2794,64	8209,45	2669,61	7068,37
45	3152,01	8869,53	3014,3	7739,93
50	3507,52	10784,89	3352,60	8750,16

Tabla 5.5: Datos de tiempo para la obtención de nivel de *ADM* y *AMD* en las topologías lineal y en malla para la red con tráfico usando valores normalizados en términos de *B*

nivel	Retardo $X = 1$	Retardo $X = 5$	Retardo $X = 10$	σ^2 con $X = 10$
5	8,50	41,20	78,33	381,87
10	14,69	71,83	138,13	674,031
15	21,01	101,87	203,03	1070,37
20	27,66	134,53	266,70	1578,50
25	33,80	164,80	328,73	1598,52
30	40,36	196,96	392,00	1988,26
35	45,12	219,37	442,11	2218,49
40	53,70	257,51	520,99	2740,80
45	60,25	288,77	585,41	3517,39
50	66,35	319,43	650,699	3867,51

Tabla 5.6: Datos de tiempo promedio en minutos requerido por nodos de diferentes niveles para la detección de cambios en la red, con la red en reposo y usando topología lineal

durará solamente 1 hora cuando hay tráfico en la red. Este resultado refleja el hecho de que cuando la red está en reposo depende del parámetro X para refrescar su información de nivel, mientras que cuando hay tráfico el nodo pasará al estado de descubrimiento de nivel si no escucha una trama de *PT* válida sin importar el valor en el parámetro X , esto también explica en hecho de que el parámetro X no tiene un efecto tan grande cuando hay tráfico en la red, en el retardo de configuración del nivel de red de los nodos como lo tiene cuando la red se encuentra en reposo.

nivel	Retardo $X = 1$	Retardo $X = 5$	Retardo $X = 10$	σ^2 con $X = 10$
5	8,26	19,78	24,44	38,77
10	14,86	35,76	42,19	53,93
15	20,91	51,43	59,78	89,56
20	27,36	67,44	78,38	113,66
25	33,38	83,19	96,89	136,29
30	40,22	99,29	115,89	147,40
35	44,96	111,61	130,05	159,99
40	53,12	132,38	154,07	213,35
45	59,20	147,04	173,40	265,50
50	65,73	162,64	191,65	317,24

Tabla 5.7: Datos de tiempo promedio en minutos requerido por nodos de diferentes niveles para la detección de cambios en la red, con la red con tráfico y usando topología lineal

5.2 Latencia de las tramas

Para medir la latencia de las tramas se utilizarán dos experimentos. En el primero se insertarán mil alarmas con una distancia de un minuto entre cada una, para los nodos de niveles múltiplos de cinco y se medirá el retraso de las mismas en función de B , utilizando la topología lineal para diferentes valores del tiempo de hibernación T en el segundo experimento se utilizará la topología en malla de 51×51 usada en el experimento anterior y se insertarán alarmas en quinientos nodos aleatorios a diferentes frecuencias para medir el efecto que esto tiene en el retardo de las tramas a diferentes niveles usando un $T = 4$.

Resultados del primer experimento:

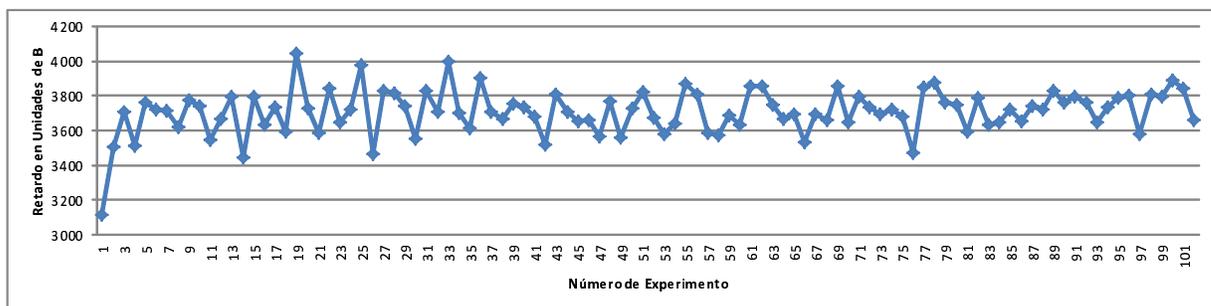


Figura 5.5: Resultado de latencia para las tramas de nivel cincuenta en algunos de los mil experimentos realizados cuando $T = 4P$

En la tabla 5.8 se resumen los resultados obtenidos para el experimento cuando se usa $T = 4P$, se puede ver que para todos los casos los retardos promedio en función de B se encuentran dentro de los rangos esperados del análisis teórico, además se puede ver

nivel	Retardo mínimo esperado	Retardo máximo esperado	Retardo promedio	σ^2
5	75	625	306,44	1623,31
10	150	1195	602,13	2616,98
15	225	1765	902,77	4127,24
20	300	2335	1206,05	6679,56
25	375	2905	1508,77	7834,38
30	450	3475	1808,75	10121,96
35	525	4045	2111,81	11284,34
40	600	4615	2413,07	14278,23
45	675	5185	2716,42	16011,45
50	750	5755	3698,78	18709,39

Tabla 5.8: Datos de tiempo promedio normalizado en B requerido por las tramas enviadas desde diferentes niveles para llegar a la base usando topología lineal, insertando una alarma por minuto en los nodos de nivel múltiplo de cinco con $T = 4P$. Dado que los datos son normalizados no tiene unidades

que conforme aumenta el nivel del nodo emisor de la alarma la varianza σ^2 del retardo aumenta, en la figura 5.5 se puede ver las diferencias en el retardo para algunos de las alarmas detectadas en nodos de nivel 50 usando $T = 4P$.

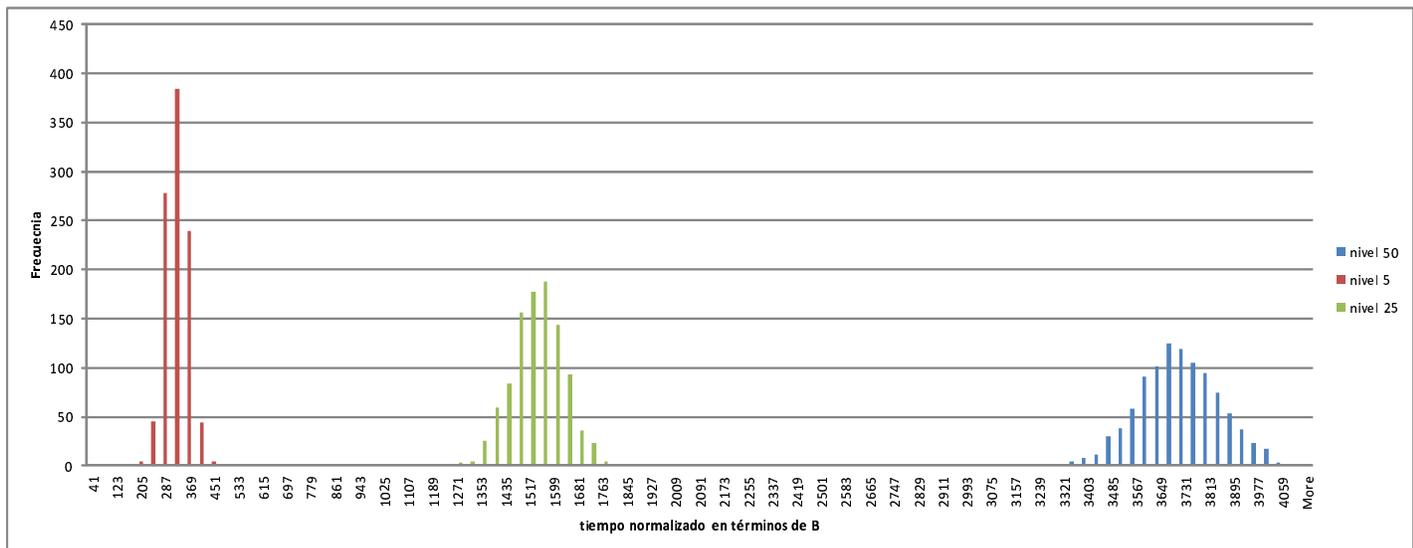


Figura 5.6: distribución de frecuencia del retardo de las alarmas generadas en nodos de niveles 5, 25 y 50 con $T = 4P$

En la figuras 5.6 se puede notar que para diferentes niveles de los nodos emisores de las alarmas la función de probabilidad en el retraso de las tramas se ve muy similar a una curva normal, en donde la varianza aumenta conforme se aumenta el nivel del nodo emisor

de la alarma, lo cual concuerda con el análisis teórico de la latencia que se expuso en la sección de diseño del protocolo en la que se modela el retardo de las tramas.

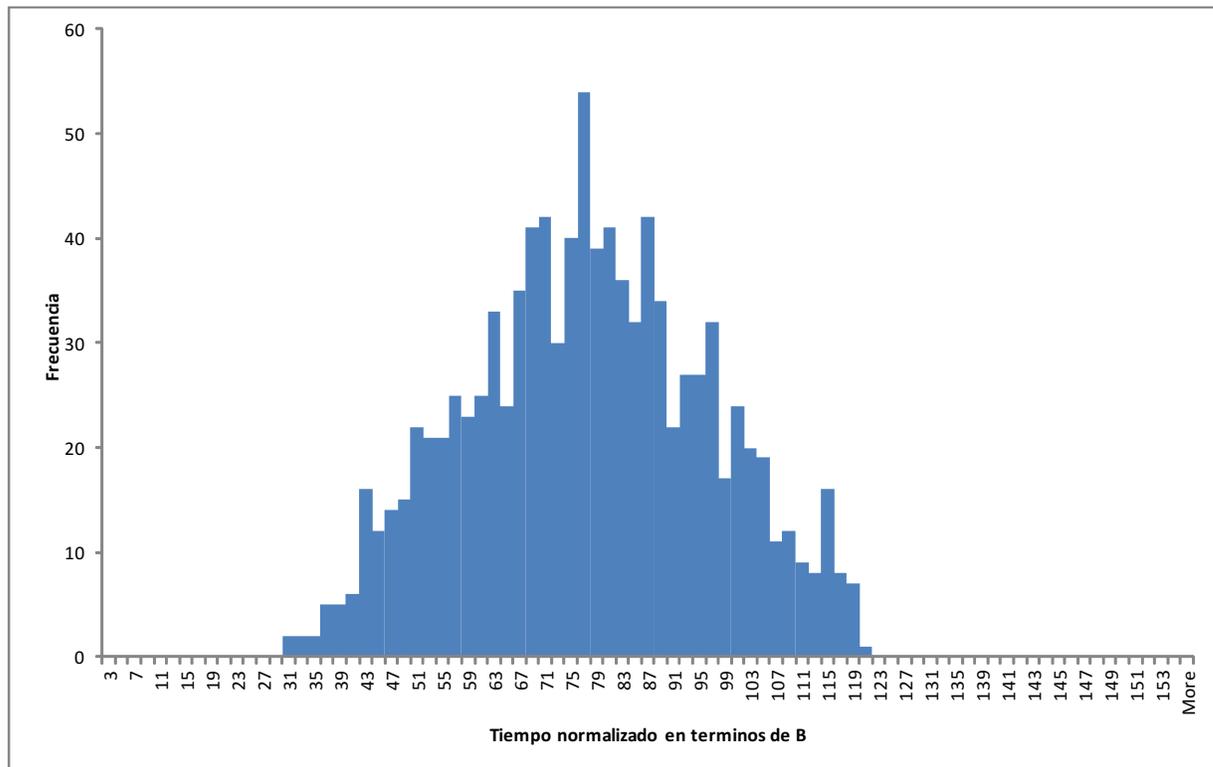


Figura 5.7: distribución de frecuencia del retardo de las alarmas para saltar de un nodo al siguiente con $T = 4P$

En la figura 5.7 se observa la distribución de frecuencia para la latencia al enviar una trama de un nivel al siguiente usando la topología lineal como referencia, dicho gráfico se puede comparar con la figura 4.15, en la que se obtenía la distribución de probabilidad teórica del retardo en el envío de una alarma de un nivel al siguiente, se puede notar que en comparación la curva obtenida en simulación se encuentra corrida hacia la derecha aproximadamente 25 unidades de B y tiene una varianza mayor que la estimada teóricamente. Esta diferencia puede deberse a simplificaciones hechas en el modelo teórico, como por ejemplo que se asume que P va a ser exactamente $11B$ lo cual no será cierto en la simulación ya que hay que tomar en cuenta el tiempo de transmisión de las tramas o por ejemplo que alguno de los nodos tuvieran que esperar por haber escuchado una trama RTS o PT de alguno de sus nodos vecinos.

En la figura 5.8 se muestran los resultados obtenidos al medir la latencia promedio para diferentes valores T desde nodos en diferentes niveles de ADM. Se observa que el retardo crece de manera aproximadamente lineal con diferentes pendientes para los diferentes niveles conforme se aumenta T .

Resultados del segundo experimento:

En la tabla 5.9 se resumen los resultados obtenidos en el segundo experimento, se puede

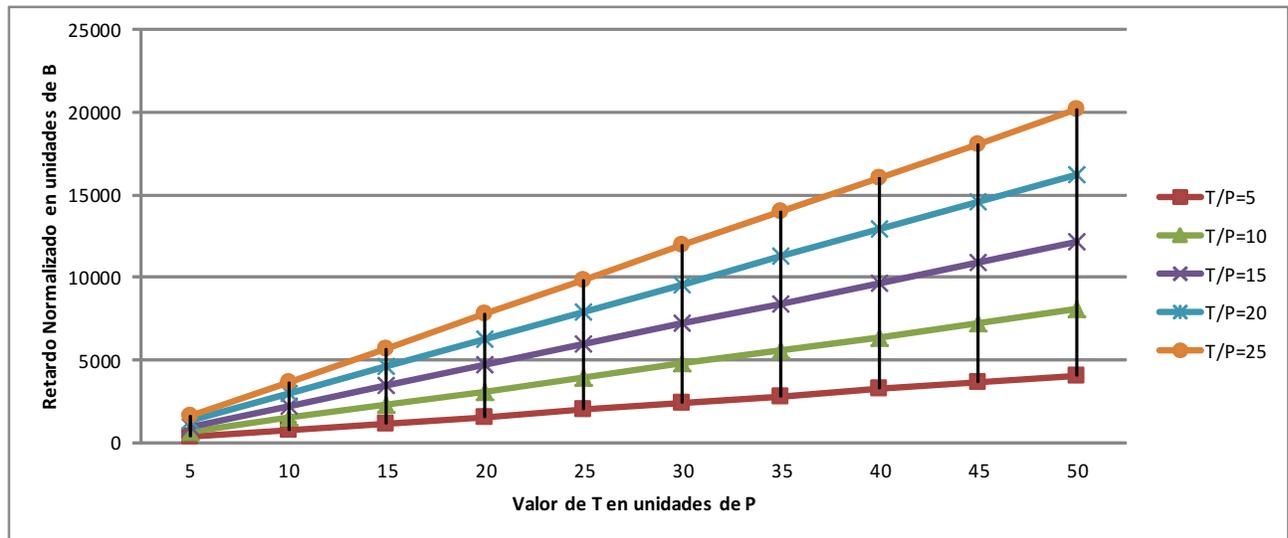


Figura 5.8: Latencia promedio usando diferentes valores de T para nodos que se encuentran en diferentes niveles de ADM

nivel	Periodo 5000B	Periodo 4000B	Periodo 3000B	Periodo 2000B	Periodo 1000B	Periodo 500B
5	282,85	325,06	333,82	396,84	490,99	673,83
10	551,27	641,81	660,41	784,89	968,41	1330,32
15	824,52	961,06	988,52	1183,20	1453,86	1990,16
20	1098,50	1280,38	1315,02	1574,07	1937,28	2663,61
25	1372,27	1599,16	1644,67	1969,54	2420,83	3329,16
30	1647,34	1918,41	1969,94	2361,63	2904,68	3996,33
35	1922,16	2238,97	2300,48	2751,85	3386,00	4655,59
40	2195,74	2558,47	2626,16	3141,38	3867,33	5324,00
45	2471,46	2878,76	2955,85	3534,90	4350,52	5993,74
50	3370,97	3924,19	4032,10	4814,06	5930,31	8165,10

Tabla 5.9: Datos de tiempo promedio normalizado en B requerido por las tramas enviadas desde diferentes niveles para llegar a la base usando una topología en malla y diferentes frecuencias para la generación de alarmas en 500 nodos aleatorios a la vez. Dado que los datos son normalizados corresponden a unidades de B y no tienen unidades

notar que la latencia de las alarmas no solamente aumenta conforme se aumenta el tráfico en la red, sino que aumenta en proporciones diferentes para los diferentes niveles de los nodos de la red, a mayor nivel mayor aumento en la latencia conforme se aumenta el tráfico, esto se ve con claridad en la figura 5.9.

Si se asume $B = 0,058s$ se tiene que para una alarma que se genere en una red con topología de malla en la que se generen alarmas en quinientos nodos aleatorios cada

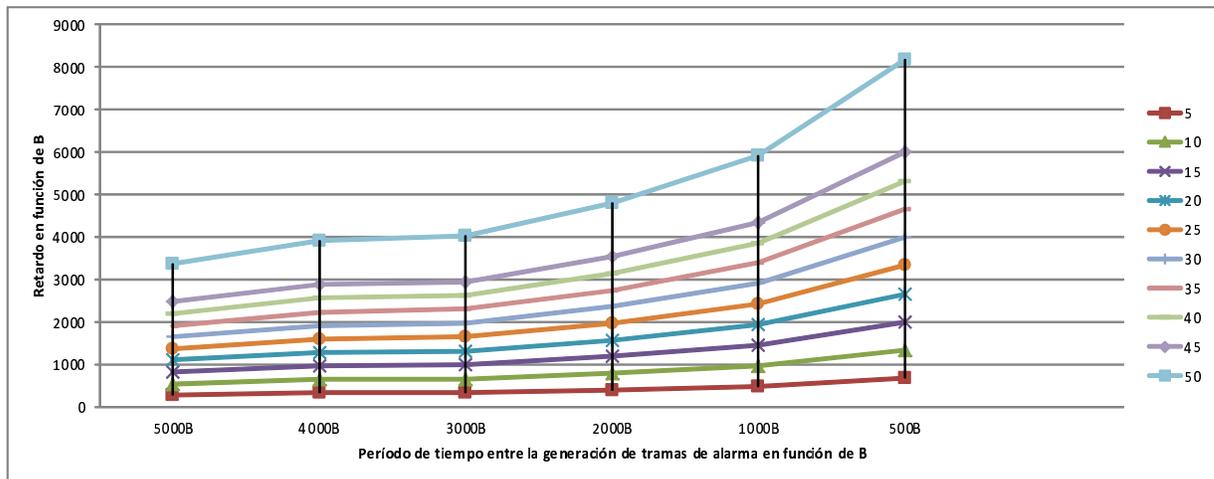


Figura 5.9: Latencia promedio normalizada en B de las alarmas en llegar a la base para nodos en diferentes niveles de ADM de la red para diferentes niveles de tráfico

veintinueve segundos, lo cual para la aplicación que estamos evaluando sería un tráfico muy alto, tendrá una latencia promedio de aproximadamente ocho minutos en llegar a la base mientras que si la alarma se genera cuando el tráfico que se está dando en la red es de quinientas alarmas en nodos aleatorios cada 290s, entonces la alarma tendrá una latencia promedio de aproximadamente tres minutos.

5.3 Consumo de potencia

Para medir el consumo de potencia se utilizan dos pruebas. En la primera se mide el consumo de potencia promedio de la red (entiéndase la suma del consumo de potencia promedio de todos los nodos en la red dividido entre el número de nodos) en reposo para dos redes con la misma configuración física, en esta prueba se usa la topología de malla de 51×51 pero usando en un caso el protocolo propuesto y en el otro los nodos de referencia conforme se aumenta el tiempo de hibernación, y en la segunda se hace manteniendo el tiempo de hibernación e insertando alarmas en 500 nodos aleatorios a diferentes frecuencias de la misma manera que se hizo en la prueba de retardo, para ver el efecto que esto tiene en el consumo de potencia. Para cada uno de los casos se repite el experimento cien veces y se reportan los resultados promediados después de cien minutos de simulación.

Para la simulación que utiliza los nodos de referencia se utilizó la interfaz Nic802154 TICC2420 de Texas Instruments implementada en la biblioteca de simulación Mixim 2.1, la cual implementa CSMA no ranurado (nonbeacon-enabled unslotted CSMA-CA) en modo competitivo.

Los parámetros utilizados en el simulador para describir el consumo de potencia de la capa física se describen en la tabla 5.1 y son los parámetros por defecto que usa Mixim

para la configuración de la capa física del nodo de referencia.

Además de los parámetros anteriormente mencionados se define la base de tiempo B en 58ms. Esto basado en los parámetros del módulo TR1000 usados también como referencia en [36]. La razón por la que se usa como referencia el transceptor TR1000 en lugar del CC2024 es porque no fué posible encontrar el dato en la hoja de datos de este último dispositivo. El resto de los parámetros de la red se describen en la tabla 5.2 con la diferencia de que para esta prueba se utilizará $X = 50$.

T/P	Potencia referencia	Potencia del protocolo propuesto	%Ahorro	%Ahorro modelado	Diferencia
5	0,0574	0,0109	81,0432	83,1854	2,1422
10	0,0574	0,0074	87,1840	90,7702	3,5862
15	0,0574	0,0059	89,6499	93,6145	3,9646
20	0,0574	0,0049	91,4135	95,1044	3,6909
30	0,0574	0,0038	93,4570	96,6423	3,1853
40	0,0574	0,0037	93,5906	97,4301	3,8395

Tabla 5.10: Tabla de resumen de los valores de potencia promedio de la red en reposo simulados usando diferentes valores de tiempo de hibernación

Asumiendo, como se aproximó en el capítulo 4, que el tiempo P es aproximadamente igual a $11B$, se tiene que la potencia promedio simulada será cómo se muestra en la tabla 5.10. Además del valor de potencia promedio, se muestran en la tabla 5.10, los valores esperados con el modelo mostrado en el apartado de diseño del protocolo 4.

Se puede ver una diferencia que fluctúa entre 2% y 3,5%. Esta diferencia puede deberse a que en el modelo no se tomaron en cuenta corrientes diferentes en los periodos de transición entre los estados de escucha, transmisión e hibernación, además de que no se tomó en cuenta el tiempo necesario para el descubrimiento de nivel que se hará periódicamente cada cincuenta veces que se pase por el estado de hibernación.

Al igual que en el experimento anterior se procede a inyectar alarmas en grupos aleatorios de quinientos nodos en la red con topología de malla

T/P	Periodo ∞	Periodo 120s	Periodo 60s	Periodo 30s	Periodo 15s	Periodo 10s
10	0,0109	0,01412	0,01963	0,0315	0,0546	0,0590
20	0,0049	0,01054	0,01620	0,0280	0,0498	0,0573
30	0,0038	0,00888	0,01464	0,0264	0,0487	0,0573
40	0,0037	0,00810	0,01373	0,0249	0,0475	0,0578

Tabla 5.11: Datos de potencia promedio en Watts para proporciones de T/P simulados al generar alarmas a diferentes frecuencias

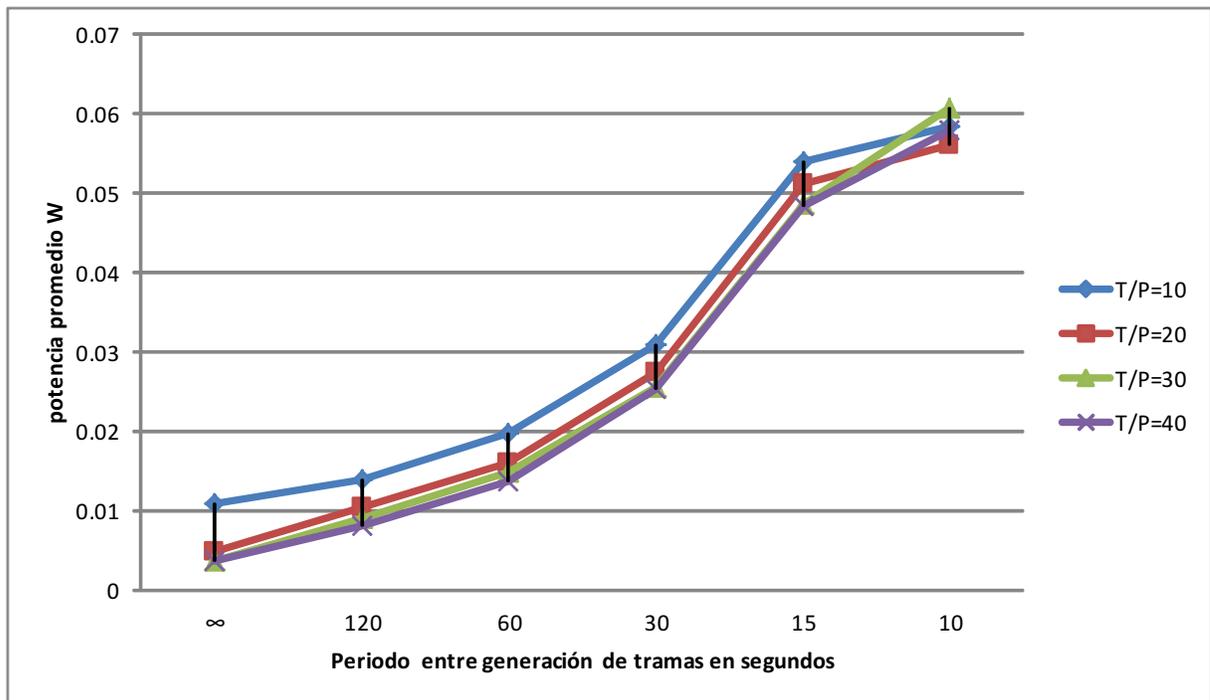


Figura 5.10: Latencia promedio normalizada en B de las alarmas en llegar a la base para nodos en diferentes niveles de ADM de la red para diferentes niveles de tráfico

En la tabla 5.11 se muestra el resultado obtenido. Se puede notar cómo el consumo de potencia crece de forma rápida conforme aumenta el tráfico en la red, este comportamiento es de esperarse debido a que conforme aumenta el tráfico en la red es necesario que los nodos mantengan la etapa de radio encendida por más tiempo para transmitir la información. Es importante apreciar que dado que la aplicación para la que pensó el protocolo propuesto asume que se va a encontrar en reposo la mayoría del tiempo el ahorro energético de la red se asume que será alto.

nivel	Potencia promedio(W) con la red en reposo	Potencia promedio(W) generando 500 alarmas cada 10 segundos
2	0,0037	0,0602
5	0,0037	0,0591
20	0,0037	0,0585
30	0,0037	0,0576
40	0,0037	0,0572
50	0,0037	0,0569

Tabla 5.12: Datos de potencia promedio en Watts para los nodos de diferentes niveles de ADM usando $T = 40$

En la tabla 5.12 se muestran los resultados de potencia promedio consumida para los

nodos de diferentes niveles de la red, se puede notar que la potencia consumida en los niveles bajos de la red cuando se generan quinientas alarmas en nodos aleatorios cada diez segundos es mayor en los nodos de niveles bajos, este resultado puede deberse a que para la topología seleccionada todos los paquetes fluyen hacia un solo sumidero de forma que todos los paquetes deben pasar por lo nodos de nivel bajo, los cuales son menos.

Capítulo 6

Conclusiones

En el presente trabajo se ha propuesto un protocolo para una aplicación específica de monitorización de variables en el bosque, de muy bajo tráfico, en donde se toma ventaja de las características de la aplicación para apagar la etapa de radio de los nodos que forman la red por periodos de tiempo que es posible configurar para mejorar el consumo energético. Se modela la relación entre el tiempo de hibernación de los nodos y el ahorro energético de los mismos con respecto de una red en la que no se apaga la etapa de transmisión de los nodos, y se modela la latencia en la entrega de las alarmas al nodo base dependiendo del tiempo de hibernación y la cantidad de saltos necesarios para que las alarmas alcancen la base. Dicho modelado provee al diseñador de la red de herramientas para configurarla de manera que cumpla con los requerimientos específicos de la aplicación se busque soportar. Con respecto del consumo de potencia en los nodos se comprueba que será inversamente proporcional al tiempo asignado en el estado hibernación cuando la red se encuentra en reposo y de relación directamente proporcional a la base de tiempo B , además se observa que consumo de potencia en la red aumenta de manera rápida conforme aumenta el tráfico en la red para aproximarse al consumo de un protocolo que mantiene su etapa de radio encendida de manera constante, por lo que se obtiene un mayor ahorro energético en aplicaciones en las que se sabe que existirá un tráfico de información bajo, además se comprueba por medio de simulación que el protocolo propuesto es capaz de autoconfigurarse para adecuarse a posibles fallas de los nodos y entregar las alarmas detectadas siempre y cuando no queden aislados de manera permanente (sin conexión a la base) o sufran un daño antes de poder entregar las tramas al siguiente nodo. Para facilitar el proceso de configuración de la red todos los tiempos del protocolo se basan en el parámetro B , el cual se define como el tiempo requerido para que el paquete más largo de la red se transmita a la distancia máxima de alcance del nodo transmisor y sea recibida por el nodo receptor, dicho parámetro depende de la capa física de la Red y deberá ser determinado para la implementación final de la misma. Con respecto al nodo de referencia Nic802154 TICC2420 de Texas Instruments el cual implementa el estándar IEEE802.15.4 con CSMA no ranurado (nonbeacon-enabled unslotted CSMA-CA) en modo competitivo, y usando wiseroute como protocolo de red, el protocolo propuesto es capaz de ahorrar aproximadamente un 93% de energía si se usa una proporción entre el periodo

de hibernación y el periodo P de 40:1 con la red en reposo. Con respecto a la latencia de las tramas, esta va a depender de factores de configuración del protocolo como la base de tiempo B y el tiempo de hibernación seleccionado. Además dependerá de factores externos al protocolo como lo son la topología de la red y la carga de la misma. Si se asume $B = 0,058s$ se tiene que para una alarma que se genere en una red con topología de malla en la que se generen alarmas en quinientos nodos aleatorios cada 29 segundos, lo cual para la aplicación que estamos evaluando sería un tráfico muy alto, tendrá una latencia promedio de dieciséis minutos en llegar a la base mientras que si la alarma se genera cuando el trafico es 500 alarmas en nodos aleatorios cada 290s, entonces la alarma tendrá una latencia promedio de 3 minutos. Lo cual cumple con el objetivo de mantener una latencia de menos de 20 minutos para alarmas generadas en el nivel cincuenta propuesto al inicio del proyecto. Para el proyecto expuesto se cumplen todos los objetivos propuestos inicialmente.

6.0.1 Trabajo Futuro

Algunas de las tareas de interés para desarrollar en trabajos futuros:

- hacer comparaciones de consumo de potencia sin usar el *ADM* y usando solamente el *AMD* para medir su impacto.
- hacer simulaciones para determinar el impacto que tiene el *ADM* en la cantidad de colisiones en los diferentes dominios de colisión usando diferentes densidades de nodos y cambiando el alcance de los mismos.
- Implementar el protocolo en un grupo de nodos físicos y compararlos contra nodos físicos que utilicen zigbee.
- Agregar seguridad al protocolo.

Bibliografía

- [1] Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans). *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pages 1–320, 2006.
- [2] A.A. Ahmed, Hongchi Shi, and Yi Shang. A survey on network protocols for wireless sensor networks. In *Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on*, pages 301–305, 2003.
- [3] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE*, 11(6):6–28, 2004.
- [4] Lichun Bao and J. J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking, MobiCom '01*, pages 210–221, New York, NY, USA, 2001. ACM. URL <http://doi.acm.org/10.1145/381677.381698>.
- [5] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. Macaw: a media access protocol for wireless lan's. *SIGCOMM Comput. Commun. Rev.*, 24(4):212–225, October 1994. URL <http://doi.acm.org/10.1145/190809.190334>.
- [6] José Joaquín Campos, Róger Villalobos, Marlen Camacho, Manuel Gómez, and Carlos Rodríguez. La tala ilegal en Costa Rica [online]. 2001 [visitado el 7 de diciembre de 2013]. URL <http://www.fao.org/forestry/12916-07ab84f47d392fcba7c7ffacd75758870.pdf>.
- [7] Ugo Maria Colesanti, Carlo Crociani, and Andrea Vitaletti. On the accuracy of omnet++ in the wireless sensornetworks domain: simulation vs. testbed. In *Proceedings of the 4th ACM workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, PE-WASUN '07*, pages 25–31, New York, NY, USA, 2007. ACM. URL <http://doi.acm.org/10.1145/1298197.1298203>.
- [8] N.M. Freris, H. Kowshik, and P.R. Kumar. Fundamentals of large sensor networks: Connectivity, capacity, clocks, and computation. *Proceedings of the IEEE*, 98(11):1828–1846, 2010.

- [9] Jimena Garbarino. Protocolos para redes inalámbricas de sensores [online, visitado el 7 de diciembre de 2013]. URL <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CDoQFjAA>
- [10] B. Gedik, Ling Liu, and P.S. Yu. Asap: An adaptive sampling approach to data collection in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 18(12):1766–1783, 2007.
- [11] S. Goel, A. Passarella, and T. Imielinski. Using buddies to live longer in a boring world [sensor network protocol]. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 5 pp.–346, 2006.
- [12] Samir Goel and Tomasz Imielinski. Prediction-based monitoring in sensor networks: taking lessons from mpeg. *SIGCOMM Comput. Commun. Rev.*, 31(5):82–98, October 2001. URL <http://doi.acm.org/10.1145/1037107.1037117>.
- [13] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670, 2002.
- [14] IEEE. Ieee standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements- part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *ANSI/IEEE Std 802.11, 1999 Edition (R2003)*, pages i–513, 2003.
- [15] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Optimal transmission radius for energy efficient broadcasting protocols in ad hoc and sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 17(6):536–547, 2006.
- [16] Texas Instruments. cc2420 2.4 ghz ieee 802.15.4 / zigbee-ready rf transceiver [online, visitado el 7 de diciembre de 2013]. URL <http://www.ti.com/lit/ds/symlink/cc2420.pdf>.
- [17] Richard Johnson. *Probabilidad y estadística para ingenieros*. Pearson, octava edition, 2012.
- [18] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. Comparative study of wireless network simulators. pages 517–523, 2008.
- [19] G. Lu, B. Krishnamachari, and C.S. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, pages 224–, 2004.

- [20] S. Mahfoudh and P. Minet. An energy efficient routing based on olsr in wireless ad hoc and sensor networks. In *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1253–1259, 2008.
- [21] memsic. Micaz wireless measurement system data sheet [online, visitado el 7 de diciembre de 2013]. URL http://www.memsic.com/userfiles/files/DataSheets/WSN/micaz_datasheet-t.pdf.
- [22] Jelena Misić, Shairmina Shafi, and Vojislav B. Misić. Performance limitations of the {MAC} layer in 802.15.4 low rate {WPAN}. *Computer Communications*, 29(13–14):2534 – 2541, 2006. URL <http://www.sciencedirect.com/science/article/pii/S014036640600051X>.
- [23] Omnet++ documentation and tutorials [online, visitado el 7 de diciembre de 2013]. URL <http://www.omnetpp.org>.
- [24] Omnet++ manual [online, visitado el 7 de diciembre de 2013]. URL <http://www.omnetpp.org/doc/omnetpp/manual/usman.html>.
- [25] Omnet++ publications [online, visitado el 7 de diciembre de 2013]. URL <http://www.omnetpp.org/publications>.
- [26] S. Park, A. Savvides, and M. B. Srivastava. SensorSim : A Simulation Framework for Sensor Networks [online]. 2001 [visitado el 7 de diciembre de 2013]. URL <http://nesl.ee.ucla.edu/projects/sensorsim>.
- [27] Sundeep Patten, Bhaskar Krishnamachari, and Ramesh Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *ACM Trans. Sen. Netw.*, 4(4):24:1–24:33, September 2008. URL <http://doi.acm.org/10.1145/1387663.1387670>.
- [28] Guangyu Pei, M. Gerla, and Tsu-Wei Chen. Fisheye state routing: a routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 1, pages 70–74 vol.1, 2000.
- [29] Stanislav Racek. General information about J-Sim [online]. 2006 [visitado el 7 de diciembre de 2013]. URL www.j-sim.zcu.cz.
- [30] N. Rahnavard, B. Vellambi, and F. Fekri. *IEEE Transactions on Wireless Communications*, 7(12):5390 – 5400, 2008. URL <http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,url,uid&db=inh&AN>

- [31] Venkatesh Rajendran, Katia Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 181–192, New York, NY, USA, 2003. ACM. URL <http://doi.acm.org/10.1145/958491.958513>.
- [32] H.S. Ramos, E.M.R. Oliveira, A. Boukerche, A.C. Frery, and A. A F Loureiro. Characterization and mitigation of the energy hole problem of many-to-one communication in wireless sensor networks. In *Computing, Networking and Communications (ICNC), 2012 International Conference on*, pages 954–958, 2012.
- [33] Jerome Rousselot. Wiseroute [online, visitado el 7 de diciembre de 2013]. URL <http://mixim.sourceforge.net/doc/MiXiM/doc/doxy/a00227.html>.
- [34] Jerome Rousselot, Jean-Dominique Decotignie, Marc Aoun, Peter van der Stok, Ramon Serna Oliver, and Gerhard Fohler. Accurate timeliness simulations for real-time wireless sensor networks. pages 476–481, 2009. URL <http://dx.doi.org/10.1109/EMS.2009.34>.
- [35] Eric Sala. Piratas en costa rica [online]. 2011 [visitado el 7 de diciembre de 2013]. URL <http://www.nacion.com/2011-12-05/Opinion/piratas-en-costa-rica.aspx>.
- [36] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *Mobile Computing, IEEE Transactions on*, 1(1):70–80, 2002.
- [37] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *Mobile Computing, IEEE Transactions on*, 1(1):70–80, 2002.
- [38] R.C. Shah and J.M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, volume 1, pages 350–355 vol.1, 2002.
- [39] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '98, pages 181–190, New York, NY, USA, 1998. ACM. URL <http://doi.acm.org/10.1145/288235.288286>.
- [40] R. Soua and P. Minet. A survey on energy efficient techniques in wireless sensor networks. In *Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP*, pages 1–9, 2011.
- [41] Sameer Sundresh, Woo Young Kim, and Gul Agha. Sensor environment and network simulator [online]. 2004 [visitado el 7 de diciembre de 2013]. URL <http://osl.cs.uiuc.edu/sens/>.
- [42] Andrew S. Tanenbaum. *Redes de computadoras 4ta edicion*. Prentice Hall, 2003.

- [43] mobile systems laboratory ucla. Global mobile information systems library [online, visitado el 7 de diciembre de 2013]. URL <http://pcl.cs.ucla.edu/projects/glomosim/>.
- [44] Gilbert Vargas Ulate. Turismo y espacios naturales protegidos en costa rica:enfrentamiento o concertación. *Rev. Ciencias Sociales Universidad de Costa Rica*, 123-124:49–78, 2009.
- [45] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM. URL <http://doi.acm.org/10.1145/958491.958512>.
- [46] M. van der Schaar and Sai Shankar N. Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms. *Wireless Communications, IEEE*, 12(4):50–58, 2005.
- [47] Qingwen Wang, Haoshan Shi, Wei Cheng, and Honggang Zhao. An adaptive probabilistic broadcasting strategy based on cross-layer design for wireless sensor network. *Chinese Journal of Sensors and Actuators*, 23(7):989 – 993, 2010. URL <http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,url,uid&db=inh&AN=>
- [48] Xiaodong Xian, Weiren Shi, and He Huang. Comparison of omnet++ and other simulator for wsn simulation. In *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, pages 1439–1443, 2008.
- [49] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pages 70–84, New York, NY, USA, 2001. ACM. URL <http://doi.acm.org/10.1145/381677.381685>.
- [50] J. Heidemann Y. Wei and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. *IEEE INFOCOM*, pages 1567–1576, 2002.
- [51] Peizhong Yi, A. Iwayemi, and Chi Zhou. Developing zigbee deployment guideline under wifi interference for smart grid applications. *Smart Grid, IEEE Transactions on*, 2(1):110–120, 2011.
- [52] Xiu-Shuang Yi, Pei-Jun Jiang, Xing-Wei Wang, and Shao-Cheng Zhang. Survey of energy-saving protocols in wireless sensor networks. In *Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on*, pages 208–211, 2011.
- [53] Rui Zhang and M.A. Labrador. Energy-aware topology control in heterogeneous wireless multi-hop networks. In *Wireless Pervasive Computing, 2007. ISWPC '07. 2nd International Symposium on*, pages–, 2007.

- [54] Wensheng Zhang and Guohong Cao. Dctc: dynamic convoy tree-based collaboration for target tracking in sensor networks. *Wireless Communications, IEEE Transactions on*, 3(5):1689–1701, 2004.

Apéndice A

Simuladores evaluados

Como se mencionó en el capítulo 3 en este apéndice se hace un muy breve resumen de los simuladores evaluados para el presente proyecto

SensorSim: [26], es un marco de simulación para redes de sensores desarrollado a partir del simulador de redes NS-2. Proporciona modelos de canales de sensores, consumo de energía, una pila de protocolos ligeros de comunicación en micro-sensores inalámbricos, creación de escenarios y simulación híbrida. El modelo de canales de sensores representa la interacción dinámica entre los sensores y el entorno físico. Cada uno de los sensores se integra un modelo de consumo de energía con varios estados, capaz de utilizar la energía de forma eficiente y prolongar la vida útil de los nodos. El simulador dejó de desarrollarse en el año 2001 y carece de documentación apropiada.

J-Sim (JavaSim): [29], es un entorno de simulación desarrollado en Java, orientado a objetos y basado en componentes. Una de las principales ventajas J-Sim es su filosofía de componentes: los módulos pueden ser añadidos y eliminados fácilmente y este aspecto suele ser interesante tanto para efectuar simulaciones de redes, como emulaciones, ya que permite incorporar uno o más dispositivos de sensores reales. J-Sim proporciona, además, soporte para distintos tipos de nodos, canales de sensores, modelos de comunicación inalámbrica, medio físico, modelos de energía y potencia.

GloMoSim (Global Mobile Information system Simulate): [43], es un entorno de simulación escalable desarrollado en C y Parsec, capaz de paralelizar la ejecución de una simulación de eventos discretos. GloMoSim se compone de una colección de bibliotecas de módulos, cada una de la cuales simula un protocolo de comunicación inalámbrica específico en la pila de protocolos. GloMoSim es ampliamente utilizado para simular redes Ad-hoc y móviles de sensores inalámbricos.

SENS (Sensor, Environment, and Network Simulator):[41] es un simulador de redes de alto nivel para aplicaciones de redes de sensores. Esta formado básicamente por tres tipos de componentes intercambiables y extensibles: Componentes físicos, de red y de aplicación. Los componentes de aplicación se utilizan para simular la ejecución en un nodo sensor, los componentes de red están orientados a simular las funciones de envío y

recepción de paquetes en el sensor y la capa física se utiliza para modelar los sensores, actuadores y modelos de potencia que interactúan con el entorno. Es posible modificar estos componentes o incluir nuevos componentes para nuevas aplicaciones, modelos de red, capacidad de sensores y entorno.

Ns-2: Es un simulador de eventos discretos orientado a objetos y una estructura modular permite la extensión de su funcionalidad. Las simulaciones se basan en una combinación de C++ y OTcl. En general, C++ se utiliza como lenguaje de programación para implementar protocolos y extender las bibliotecas con nuevas funcionalidades. OTcl se utiliza para crear y controlar en entorno de simulación, incluyendo la selección de los datos obtenidos. Ns-2 como simulador de redes de sensores, es una modificación de los módulos orientados a movilidad y redes ad hoc mediante pequeñas extensiones. Ofrece soporte para algunas características peculiares en redes de sensores como hardware limitado, modelos de potencia e incluso ha sido desarrollada una extensión que permite que fenómenos del entorno de simulación actúen como gatillos en la simulación.

OMNeT++: [23] Desarrollada en C++ proporciona una herramienta para la simulación de eventos discretos en el ámbito académico que permite modelar redes de comunicación, multiprocesadores y otros tipos de sistemas paralelos o distribuidos. OMNeT++ es de código abierto. Pretende cubrir el vacío entre herramientas de estilo de código abierto y orientadas a la investigación (como Ns-2) y herramientas comerciales de alto coste como OPNeT (OPNET Technologies, Inc.). OMNeT++ proporciona una estructura extensible, modular, orientada a componentes y jerárquica. Los componentes y módulos están programados en C++ y los desarrolladores pueden ampliar la funcionalidad de las bibliotecas del núcleo de simulación y utilidades proporcionadas para generación de números aleatorios, generación de estadísticas, topologías, etc. Se proporciona un lenguaje llamado NED (Network Descripción) utilizado para ensamblar componentes individuales en nuevos componentes y modelos. Omnet++ ha sido comparado con NS2 y se han recogido métricas que indican que emplea menos memoria y un menor tiempo de ejecución [48]. También se ha evaluado la exactitud de los resultados obtenidos con Omnet++ en simulaciones de WSN[7], observándose que cuanto mayor es la frecuencia de muestreo, la cantidad de eventos que debe manejar el nodo aumenta fuertemente. A una frecuencia de muestreo de 5 ms, los resultados de las simulaciones muestran una distancia significativa respecto de los resultados obtenidos en un campo de pruebas real. A partir de una frecuencia de 50 ms, esta distancia es menos evidente, y a partir de los 500 ms es despreciable. Posee documentación detallada [23] lo cual hace más sencilla su adopción y generación de nuevas bibliotecas, además de contar con una extensiones y artículos publicados que muestran ejemplos de uso. Aparte de los brevemente mencionados existe una gran variedad de simuladores de redes. En [18] se comparan con detalle ventajas y desventajas de diferentes simuladores. Por todo lo anterior se selecciona Omnet++ para el desarrollo de la presente investigación.

Ambiente de desarrollo

El IDE (ambiente integrado de desarrollo) de Omnet++ está basado en la plataforma Eclipse[24], y la extiende con editores, vistas y asistentes. Este IDE provee una Perspectiva de Simulación (figura 4.3) para trabajar con los tipos de archivo de Omnet++, que deben ser incluidos en un proyecto de simulación.

Definición de un módulo simple

Como se ha dicho, la simulación ejecuta módulos simples. Un módulo simple es una clase C++ que debe extender una clase módulo simple de base, y redefinir su comportamiento. La clase debe registrarse con Omnet++. Los módulos simples son instanciados por el kernel de simulación, por ello la firma del constructor debe tener una forma predefinida. Debe seguirse la siguiente convención de inicialización y finalización:

constructor: El constructor es invocado para crear el módulo, como parte de la inicialización del módulo propiamente dicho. En ese momento, todo el modelo está siendo construido por lo que no pueden hacerse muchas cosas desde el constructor. Por convención, en el constructor se asignan todas las variables puntero de la clase del módulo a NULL.

inicializador: El método `initialize()` es invocado justo antes de comenzar la ejecución de la simulación, cuando todo el modelo está listo. Por convención, en el método `initialize()` se realizan todas las tareas de inicialización: leer parámetros, inicializar variables, reservar memoria, crear e inicializar timers.

destructor: El destructor siempre se llama al final, sin importar cómo terminó la simulación. Se puede asumir que, en ese momento, el modelo de simulación está casi destruido por completo. Por convención en el destructor se libera la memoria reservada, se eliminan los temporizadores creados (con la función `cancelAndDelete(msg)`).

finalizador: El método `finalize()` sirve para grabar estadísticas. Solamente se invoca cuando la simulación ha terminado normalmente. Si la simulación se detiene por error, no sería invocado. Por convención, en este método se graban estadísticas, y no se deben liberar objetos ni ejecutar tareas de limpieza. Para definir el comportamiento dinámico del módulo simple, se debe redefinir el método `handleMessage`, invocado con un mensaje como parámetro cada vez que el módulo recibe un mensaje. Se espera que el método procese el mensaje, y retorne. Durante la ejecución de este método no transcurre tiempo de simulación, sólo entre llamadas. La idea es que con cada evento (llegada de mensaje) se invoca una función definida en el módulo. Dentro de la función pueden utilizarse otras funciones predefinidas para enviar mensajes a otros módulos, programar un evento (enviar un mensaje a sí mismo), y borrar un evento programado. Los modelos de capas de protocolos tienden a compartir una estructura similar, porque reaccionan a tres tipos de eventos fundamentales: mensajes que llegan de capas superiores (la aplicación), mensajes que llegan de capas inferiores (la red) y varios temporizadores (auto-mensajes). Gene-

```
1 class FooProtocol : public cSimpleModule
2 {
3     protected:
4         //state variables
5         //...
6         virtual void processMsgFromHigherLayer (cMessage *packet);
7         virtual void processMsgFromLowerLayer (FooPacket *packet);
8         virtual void processtimer (cMessage *timer);
9         virtual void initialize ();
10        virtual void handleMessage (cMessage *msg);
11 };
```

Figura A.1: Ilustración flujo de mensajes en omnet

ralmente, surge el patrón de modelado que se ilustra en el fragmento de código que se muestra en la figura A.1:

Simulación

Una vez que se han escrito los archivos fuentes se compila el proyecto obteniendo un ejecutable. La simulación se realiza corriendo el ejecutable generado, y puede ser lanzada desde el IDE. Para ello, se define una configuración de ejecución, especificando algunos detalles como los mostrados en la figura A.2:

- **Directorio de trabajo:** Es el directorio base del cual dependen otras configuraciones.
- **Ejecutable:** la ruta del ejecutable de la simulación, relativa al directorio del espacio de trabajo
- **Archivos de inicialización:** archivos que contienen configuraciones de parámetros modificables de la red y los módulos

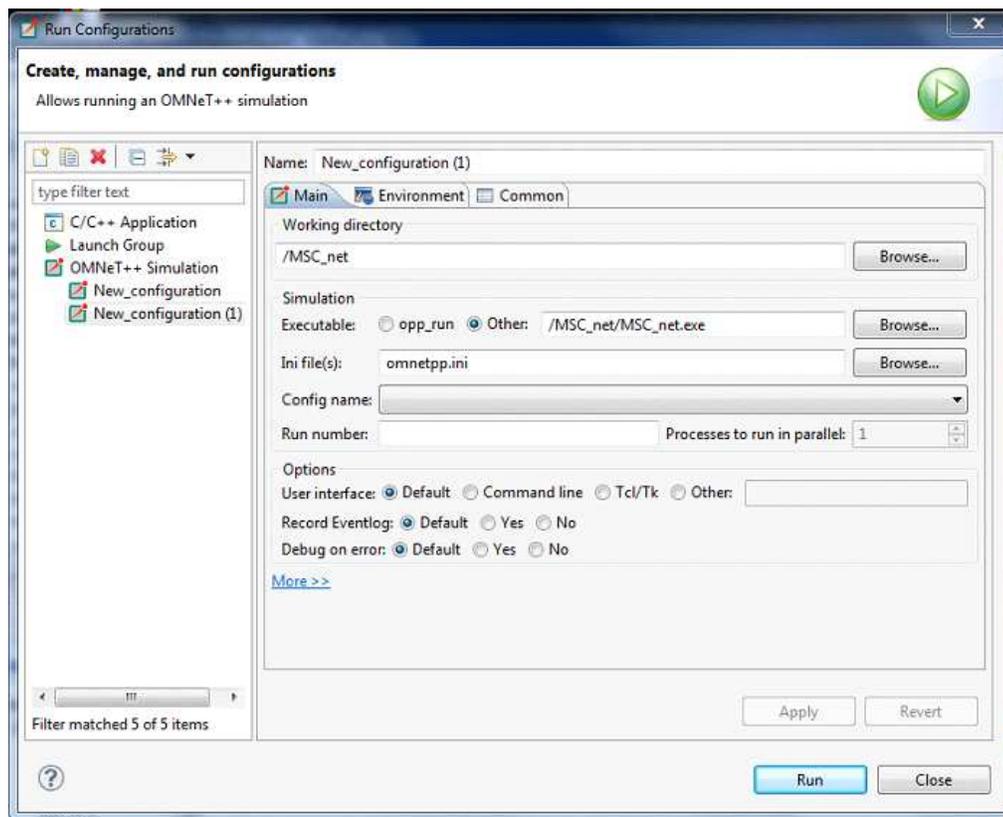


Figura A.2: Configuración de simulación en omnet

Apéndice B

Glosario

- *ACK*: Trama de acknowledge.
- *AMD*: Algoritmo de enrutamiento de mezcla de datos.
- *ADM*: Algoritmo de enrutamiento de datos mezclados.
- *B*: Tiempo necesario para transmitir la trama más larga del protocolo y que está sea escuchada por un nodo colocado al límite de la distancia máxima permitida por la etapa de radio de los nodos.
- *BER*: Bit Error Ratio (BER) traducido como Tasa de error de bit, es en telecomunicaciones, el número de bits o bloques. incorrectamente recibidos, con respecto al total de bits o bloques enviados durante un intervalo especificado de tiempo.
- *CHECK_SUM*: Es un código de detección de errores usado frecuentemente en redes digitales y en dispositivos de almacenamiento para detectar cambios accidentales en los datos. En el presente trabajo se usa simplemente la suma de los bits en el campo de la alarma y se usan los últimos 8 bits para chequear que el dato recibido es correcto.
- *CSMA*: Protocolo de acceso múltiple con detección de portadora; es un protocolo en el que cuando una estación tiene datos para transmitir primero escucha el canal para saber si otra estación está transmitiendo en ese momento. Si el canal está ocupado la estación espera hasta que el canal se desocupe. Cuando la estación detecta un canal inactivo transmite la trama; si ocurre una colisión la estación espera una cantidad aleatoria de tiempo y comienza de nuevo.
- *CTS*: Trama de apertura de canal usada para evitar el problema de estación oculta.
- *FDM*: Método para la multiplexación del canal de transmisión en el que se asigna a cada usuario un rango de frecuencia independiente de la del resto de los usuarios
- *K*: Es un parámetro del protocolo propuesto y se define como peor tiempo posible de retraso para el proceso de transmisión de una alarma de un nodo al siguiente sin tomar en cuenta el tiempo de hibernación y asumiendo que no hay colisiones
- *Limite_de_mezcla*: Es un parámetro de la red propuesta que define la cantidad máxima de tramas de alarma que se pueden transmitir en un paquete del *ADM*.
- *MAC*: Capa de control de acceso al medio del modelo de referencia OSI.
- *Max_cluster_level*: Es un parámetro de la red propuesta usado para definir las

cabezas de conglomerado. Se usarán los nodos de nivel de *ADM* múltiplo de $Max_cluster_level \times 2$ como cabezas de conglomerado

- *MPEG*: El Moving Picture Experts Group (MPEG) es un Grupo de Trabajo de expertos que se formó por ISO y IEC para establecer estandares para el audio y la transmisión video. Los estándares desarrollados por este grupo han tomado tambien el nombre de *MPEG*.
- *OSI*: Modelo de referencia usado en el desarrollo de protocolos de comunicación "Open Systems Interconnection model".
- *P*: Es un parámetro del protocolo propuesto y representa el tiempo desde que se sale del estado de hibernación hasta que se regresa al estado de hibernación, para efectos del los tiempos usados como referencia se define como igual a $11B$.
- *PT*: Trama de petición de trama usada en el protocolo propuesto.
- *RTS*: Trama de petición de canal usada para evitar el problema de estación oculta.
- *T*: Es un parámetro del protocolo propuesto y representa el tiempo de hibernación.
- *TDM*: Método para la multiplexación del canal de trasmisión en el que se asigna a cada usuario una partición o segmento de tiempo independiente del de el resto de los usuarios, evitando de esta manera las colisiones de datos.
- *X*: Es un parámetro del protocolo propuesto e indica cuantas veces continuas pasa un nodo por el estado de hibernación antes de pasar al estado de descubrimiento de nivel para reconfigurar su nivel de pertenencia.